LATEX for forfattere

en introduktion til LATEX og IMFUFA-LATEX

Larsen, Jørgen

Publication date:
1999

Document Version
Også kaldet Forlagets PDF

Citation for published version (APA):
Larsen, J. (1999). *LATEX for forfattere: en introduktion til LATEX og IMFUFA-LATEX*. Roskilde Universitet. Tekster fra IMFUFA Nr. 366 http://milne.ruc.dk/ImfufaTekster/pdf/366.pdf

TEKST NR 366 1999

LaTeX FOR FORFATTERE

En introduktion til LaTeX og IMFUFA-LaTeX

Jørgen Larsen

Maj 1999

TEKSTER fra

IMFUFA ROSKILDE UNIVERSITETSCENTER
INSTITUT FOR STUDIET AF MATEMATIK OG FYSIK SAMT DERES FUNKTIONER I UNDERVISNING, FORSKNING OG ANVENDELSER

IMFUFA · Roskilde Universitetscenter · Postboks 260 · DK-4000 Roskilde
Jørgen Larsen: LaTeX FOR FORFATTERE. En introduktion til LaTeX og IMFUFA-LaTeX.
IMFUFA tekst nr. 366/1999 118 sider ISSN 0106-6242

I gamle dage talte man om bogtrykker*kunsten*, og det at fremstille trykt materiale var noget der var forbeholdt dem der havde forstand på det. I vore dage forventer enhver der har lært at tænde for en computer at kunne fremstille professionelt udseende tekster sådan helt uden videre.

Den skribent der med computerens hjælp og uden den store indsigt i typografiens mysterier ønsker at fremstille pæne, velafbalancerede tekster hvor layoutet medvirker til formidlingen af indholdet, kan næppe vælge noget bedre redskab end LaTeX – der udtales [la·tæk].

IMFUFAs lærere og studerende har anvendt LaTeX i lang tid. De første år gav det anledning til gråd og tænders gnidsel, men efterhånden er det indoptaget som en del af stedets ånd. Det kunne se ud som om de nytilkomne lærer kunsten af de ældre i gårde, men ved nærmere eftersyn tyder noget dog på at det undertiden er de *dårligere* vaner der går i arv.

Nærværende skrift er udarbejdet på baggrund af mangeårigt virke som stedlig LaTeX-vedligeholder og -vejleder. Det indeholder svarene på en hel del spørgsmål der er stillet i tidens løb, og også på mange der burde være stillet.

IMFUFA-LaTeX er et sæt dokumentklasser der er udviklet som supplement til Standard LaTeX, blandt andet med det formål at gøre livet lidt lettere for danskskrivende LaTeX-brugere. IMFUFA-LaTeX kan findes på adressen `http://dirac.ruc.dk/imfufalatex/`

Store dele af det der står i teksten forudsætter at man har en temmelig ajourført LaTeX-installation; de afgørende elementer er

- `LaTeX2e <1998/06/01>`
- `ecfonts version 1.0`
- `babel v3.6s 1999/04/13`
- `IMFUFA LaTeX version 2.4`

Indhold

Kapitel 1

Fortale

LaTeX er et hjælpemiddel der kan benyttes i den proces hvor en forfatters produkt, en tekst, overgår fra at befinde sig inde i forfatterens hoved til at befinde sig på papir i trykt tilstand. Inden vi præsenterer LaTeX kan det derfor være nyttigt at gennemgå denne omformningsproces lidt nøjere, især som den måske foregik for halvandet hundrede år siden. Man kan tænke sig at der medvirker forskellige aktører (nogle af dem kan meget vel være den samme person):

- *Forfatteren* har to funktioner:
  - Forfatteren bestemmer først og fremmest tekstens *indhold*, og derudover også tekstens *opbygning*, dvs. tekstens opdeling i sætninger, afsnit, kapitler osv. – Dette krævende kreative arbejde interesserer os ikke her.
  - Forfatteren nedskriver sin tekst som *manuskript*.

    Ordet manuskript betyder direkte oversat *noget der er skrevet i hånden*, og vores forfatter for halvandet hundrede år siden skrev netop i hånden, med blyant eller med pen og blæk. Manuskriptet har ofte set forfærdeligt ud, fyldt af rettelser og tilføjelse, og med sære tegn og markeringer der skulle vise at et ord skulle være fremhævet, eller at der skulle begynde et nyt afsnit eller kapitel eller lignende.
- *Designeren* eller *tilrettelæggeren* (for halvandet hundrede år siden var det bogtrykkeren) fastsætter de overordnede retningslinjer for hvordan den trykte tekst skal se ud. Der er tale om afgørelser om blandt andet papirformatet, marginernes størrelse, skriftsnittet (skal man bruge antikva eller måske en blokskrift), skriftgraden (dvs. skriftens størrelse), overskrifternes udformning, sidenumrenes placering, udformningen af opstillinger med punkter og underpunkter, osv. osv.
- *Sætteren* sætter forfatterens tekst efter designerens anvisninger.

  Sætteren sørger for at en række typografiske (og æstetiske) retningslinjer bliver overholdt, for eksempel skal bogstaverne i et ord have den rette indbyrdes afstand, mellemrummene mellem ordene skal have den rigtige

størrelse, der skal være lige venstre- og højremargin, ord skal deles rigtigt, plus adskilligt andet. Derudover gælder der en række særlige regler for matematiske formler.

Det at sætteren *sætter* teksten, skulle dengang forstås helt bogstaveligt: han tog metaltyper en ad gangen og placerede dem, *satte* dem, i en stor ramme der skulle komme til at svare til en tekstside. (Typerne opbevaredes i en slags lille reol, en sættekasse, hvor hvert bogstav havde sit eget rum, de »store« bogstaver i den øverste del af kassen og de »små« bogstaver i den nederste del; deraf kommer de engelske betegnelser 'uppercase' og 'lowercase' for »store« og »små« bogstaver.)

- *Trykkeren* fremstiller papiraftryk af den sats som sætterne har lavet.
- *Korrekturlæseren* læser et prøvetryk igennem for at finde trykfejl. Forfatteren læser også prøvetrykket igennem, men det er mere for at finde tankefejl. Derefter rettes fejlene.

I vore dage er nogle af disse funktioner overtaget af computeren. I TEX-verdenen er det sådan at *sætterens* opgave er overtaget af computeren. Forfatteren og designeren er stadig mennesker, men nu kan de ikke længere gå hen og sludre med sætteren og fortælle hvad de der mærkelige krusseduller på side 17 egentlig betyder; i stedet må de betjener de sig af et standardiseret sprog til typografiske markeringer; LATEX er et sådant sprog. Når forfatteren arbejder på sit manuskript, som nu skrives på computer, skal han eller hun altså stadig markere hvis der skal ske noget særligt, f.eks. at et ord skal skrives med fed skrift, blot gøres det nu ikke mere med en eller anden slags understregning, men med en bestemt LATEX-kommando. På samme måde er designerens instruktioner til »sætteren« skrevet med LATEX-kommandoer.

Resultatet af »sætteren«s arbejde har man ikke umiddelbart den store glæde af, det er nemlig bare en (ganske ulæselig) fil, en såkaldt *dvi-fil*. Man har så brug for et andet computerprogram (en *viewer*) som kan vise den satte tekst på skærmen, eller et tredje program som kan skrive teksten ud på en printer.

Desforuden skal forfatteren jo fremstille en fil med manuskriptet (incl. LATEX-kommandoer), og til det brug har han/hun brug for en simpel tekst-editor.

Efterhånden er der ved at være så mange programmer i spil at man også har brug for et program der kan hjælpe brugeren med at holde styr på de andre programmer. Et sådant program, hvis grad af sofistikerethed er meget afhængig af hvilket operativsystem[1] computeren benytter sig af, er typisk bygget op omkring en hensigtsmæssig tekst-editor hvor man ved hjælp af simple tastekombinationer eller fra menuer kan aktivere alle de øvrige programmer.

[1]så som MS-DOS, Windows 98, Linux ...

Kapitel 2

Enkle tekster uden matematik

Dette kapitel er en blid indøvelse i de grundlæggende LaTeX-tankegange i forbindelse med fremstilling af almindelige tekster uden matematiske formler. Der kommer mange eksempler, og de er for det meste skrevet i de særlige tospaltede eksempel-kasser hvor der til højre står hvad man skal skrive (input-teksten), og til venstre hvad der kommer ud af det.

Vi begynder med at vise hvad der skal skrives for at fremstille den meget enkle tekst »Hej med dig!«:

Hej med dig!	`\documentclass{notat}`
	`\begin{document}`
	`  Hej med dig!`
	`\end{document}`

EKSEMPEL 2.1

LaTeX-tekster skal *altid* begynde med en dokumentklasseerklæring; *dokumentklassen* fastlægger det layout der skal benyttes til den type tryksag man ønsker at fremstille. Efter dokumentklasseerklæringen følger selve teksten der altid skal omgives af `\begin{document}` og `\end{document}`. I IMFUFA-LaTeX findes blandt andet dokumentklasserne notat, bog, artikel og rapport; der kommer mere om dokumentklasser på side 15.

Trods sin enkelhed illustrerer eksemplet et par væsentlige ting. Der optræder særlige »ord« der begynder med en *backslash* (dvs. tegnet \); det er *kommandoer*, dvs. instruktioner til LaTeX. Backslash-tegnet betyder altid at det der kommer umiddelbart efter, skal opfattes på en særlig måde, nemlig som en såkaldt *kommando*, dvs. en instruktion til LaTeX. Allerede i første linje er der en kommando, nemlig `\documentclass`. Kommandoer kan have krævede og/eller valgfrie *argumenter*; i eksemplet har kommandoen `\documentclass` argumentet `notat`; krævede argumenter skal omsluttes af `{ ... }`.

Eksemplet indeholder også parret `\begin{document}` `\end{document}`. LaTeX-dokumenter indeholder normalt et hav af sådanne `\begin{`⟨*noget*⟩`}` `\end{`⟨*noget*⟩`}` konstruktioner; deres funktion er at afgrænse et område af input-teksten hvor der skal gælde ⟨*noget*⟩; i særdeleshed afgrænser `\begin{document}` `\end{document}` det område hvor LaTeX-reglerne gælder.

For `\begin{...}` `\end{...}`-konstruktioner gælder den simple regel at ethvert `\begin{...}` *skal* have et modsvarende `\end{...}` (og omvendt).

I de de følgende eksempler vil vi kun gengive den interessante del af input-teksten. Det er altså til stadighed underforstået at input-teksten skal omsluttes af `\begin{document}` `\end{document}`, og at der skal være en dokumentklasseerklæring allerførst.

Vi udvider det første eksempel lidt:

EKSEMPEL 2.2

Hej med dig! Du er vist en *værre* en, er du ikke også?	`Hej med dig!`
	`Du er vist en \emph{værre} en,`
	`er du ikke også?`

Her er brugt kommandoen `\emph` der sætter sit argument (`værre`) i kursiv. Kommandoen benyttes når man vil *fremhæve* (engelsk: emphasize) en tekstdel, og den er et godt eksempel på arbejdsdelingsfilosofien i LaTeX: forfatteren bestemmer *at* en bestemt tekst-del skal fremhæves, og designeren/layouteren (mere præcist: dokumentklassen) bestemmer *hvordan* det skal ske.

Man kan endvidere se at det ikke har nogen betydning for slutresultatet om der er et eller flere mellemrum mellem ordene i input-teksten. Det betyder heller ikke noget at der er et enkelt linjeskift, men se så her:

EKSEMPEL 2.3

Hej med dig!
Du er vist en *værre* en, er du ikke også?
	`Hej med dig!`
	
	`Du er vist en \emph{værre} en,`
	`er du ikke også?`

En blank linje betyder *begynd på et nyt afsnit*, hvilket i den resulterende tekst markeres ved linjeskift plus indrykning (se dog side 67). I stedet for en blank linje kan man benytte kommandoen `\par`:

EKSEMPEL 2.4

Hej med dig! Du er vist en *værre* en, er du ikke også?	`Hej med dig!\par Du er vist en` `\emph{værre} en, er du ikke også?`

Hvis man i teksten vil gå over til noget lidt andet og gerne vil markere det anderledes end det almindelige »nyt afsnit«, kan man lave et *mellemslag* med kommandoen `\bpar`:

EKSEMPEL 2.5

Hej med dig! Du er vist en *værre* en, er du ikke også?	`Hej med dig!\bpar Du er vist en` `\emph{værre} en, er du ikke også?`

Enkle LaTeX-kommandoer

LaTeX-kommandoer har næsten altid navne af formen en backslash (\) efterfulgt af et eller flere amerikanske bogstaver (A... Z a... z), og de kan have nul eller flere

argumenter der skrives i { }. Vi har allerede set kommandoen `\emph` der skriver sit argument med fremhævet skrift. Et andet eksempel er kommandoen `\textqm` der sætter anførselstegn ('quotation marks') om sit argument. Eksempel:

En »køn« historie! Hun sagde: »En ›køn‹ historie!«	`En \textqm{køn} historie! Hun sagde: \textqm{En \textqm{køn} historie!}`

EKSEMPEL 2.6

(Brugeren kan ændre arten af anførselstegn, f.eks. til „ “, se nr. 8 på side 87).

Endnu et eksempel på en kommando med ét argument er kommandoen `\footnote` der skriver sit argument som en fodnote:

Nogle forfattere er glade for fodnoter,[a] men for mange fodnoter gør teksten svær at læse. [a]En god ting kan dog også overdrives.	`Nogle forfattere er glade for fodnoter,\footnote{En god ting kan dog også overdrives.} men for mange fodnoter gør teksten svær at læse.`

EKSEMPEL 2.7

Nogle kommandoer har flere argumenter; kommandoen `\nicefrac` har to:

$^1/_2$ kg saftige ribs og $^1/_4$ kg fuldmodne hindbær vrides. Heraf fås $3^1/_2$ dl saft.	`\nicefrac{1}{2} kg saftige ribs og \nicefrac{1}{4} kg fuldmodne hindbær vrides. Heraf fås 3\nicefrac{1}{2} dl saft.`

EKSEMPEL 2.8

Man kan sagtens have kommandoer med nul argumenter, for eksempel bliver LaTeX-logoet fremstillet af kommandoen `\LaTeX`:

LaTeXer nemt.	`\LaTeX er nemt.`

EKSEMPEL 2.9

Dette eksempel viser at LaTeX måske alligevel ikke er helt så nemt, for hvorfor er der nu ikke noget mellemrum mellem »LaTeX« og »er«? Det kommer sig af at når LaTeX-fortolkeren støder på tegnet \, skifter den fra en »læs ord«-tilstand til en »læs kommando«-tilstand, og i »læs kommando«-tilstanden er mellemrumstegnets betydning den at det kan bruges til at afslutte et kommandonavn, men derudover giver det ikke noget synligt resultat. Og hvad gør man så? Ja, her vises tre forskellige løsningsmuligheder; til lejligheden er mellemrumstegnet i input-teksten vist som ␣:

LaTeX er nemt.	`{\LaTeX}␣er␣nemt.`
LaTeX er nemt.	`\LaTeX{}␣er␣nemt.`
LaTeX er nemt.	`\LaTeX\␣er␣nemt.`

EKSEMPEL 2.10

I den første løsning er der sat TeX-parenteser ({ }) omkring `\LaTeX` for at angive at denne kommando skal behandles for sig selv; i den anden løsning bruges det tomme tegn {} til at markere afslutningen på kommandonavnet; i den tredje løsning benyttes kommandoen »backslash-mellemrum« (\␣) der giver et almindeligt ord-mellemrum.

Særlige tegn

Visse tegn, nemlig \ % { } $ " & # _ ^ ~, har en helt speciel betydning:

\ en *backslash* bruges til at markere LaTeX-kommandoer.

Teksttegnet \ frembringes med kommandoen `\textbackslash`.

% *procenttegn* er et kommentar-tegn. Tegnet selv og hele resten af linjen ignoreres.

Teksttegnet % frembringes med kommandoen `\%`.

{ } *krøllede parenteser* bruges om argumenter til kommandoer, og de bruges også som parenteser i forhold til LaTeX-kommandosproget, dvs. de kan afgrænse rækkevidden af definitioner.

Teksttegnene { og } frembringes med kommandoerne \{ og \} eller `\textbraceleft` og `\textbraceright`.

$ *dollartegn* skifter frem og tilbage mellem LaTeXs tekst- og matematik-tilstand.

Teksttegnet $ frembringes med kommandoen `\$`.

" *dobbelt anførselstegn* er et specialtegn i babel-pakken (se side 36) og må *ikke* bruges som anførselstegn; dels er det grimt, dels virker det ikke altid som man skulle forvente (se f.eks. 6.4 på side 68).

& *og-tegn* bruges som tabulatortegn i tabeller og arrays.

Teksttegnet & frembringes med kommandoen `\&`.

nummertegn bruges i forbindelse med definition af kommandoer.

Teksttegnet # frembringes med kommandoen `\#`.

_ *understreg-tegn* bruges i matematik-tilstand til »fodtegn« (som i x_n).

Teksttegnet _ frembringes med kommandoen `\_`.

^ *circumflex-tegn* bruges i matematik-tilstand til »toptegn« (som i x^n).

Teksttegnet ^ frembringes med kommandoen `\textasciicircum`. Hvis man vil sætte en circumflex-accent over et bogstav, gøres det med kommandoen `\^`, se Tabel 2.2.

~ *tilde* bruges til at angive et fast mellemrum, dvs. et ordmellemrum hvor der ikke må forekomme linjeskift (se Eksempel 6.7 side 68).

Teksttegnet ~ frembringes med kommandoen `\textasciitilde`. Hvis man vil sætte en tilde over et bogstav, gøres det med kommandoen `\~`, se Tabel 2.2.

æ	\ae	Æ	\AE
ø	\o	Ø	\O
å	\aa	Å	\AA
ß	\ss	SS	\SS
œ	\oe	Œ	\OE
ł	\l	Ł	\L
ı	\i	J	\j
ŋ	\ng	Ŋ	\NG
ð	\dh	Ð	\DH
þ	\th	Þ	\TH

Tabel 2.1 Særlige bogstaver og deres LaTeX-kommandoer.

ù	\`{u}	gravis	ú	\'{u}	akut
û	\^{u}	circumfleks	ũ	\~{u}	tilde
ü	\"{u}	Umlaut	ŭ̃	\H{u}	
ŭ	\u{u}	brevis	ŭ	\v{u}	háček
u̇	\.{u}		ụ	\d{u}	
ṵ	\c{u}	cedille	ṵ	\k{u}	ogonek
ū	\={u}	macron	ṵ	\b{u}	
ů	\r{u}	bolle			

Tabel 2.2 Accenter og andre diakritiske tegn, anvendt på bogstavet u.

Særlige bogstaver

TeX og LaTeX blev oprindeligt udviklet i USA, og (blandt andet) derfor anerkendte de oprindelige udgaver kun de to gange 26 bogstaver ABCD...XYZ og abcd...xyz som rigtige bogstaver. Sidenhen er LaTeX i høj grad også blevet et europæisk projekt, og udmærkede bogstaver som æ ø å ř é ä ß ð þ er kommet ind i varmen.

Hvis man benytter en fornuftig LaTeX-installation, kan man som hovedregel gå ud fra at de bogstaver man kan skrive på tastaturet, også vil optræde på den forventede måde i den resulterende tekst. Imidlertid er det jo ikke alle tastaturer der har f.eks. en ð-tast, men så er det muligt at angive bogstaverne ved hjælp af deres LaTeX-kommandoer, se Tabel 2.1.

Derudover findes en lang række specialtegn, så som № (`\numero`), ‰ (`\promille`), € (`\euro`), ℃ (`\celsius`), og ♪ (`\musicalnote`).

Accenter og andre diakritiske tegn

Der er LaTeX-kommandoer til alskens accenter og andre diakritiske tegn; dette vises i Tabel 2.2 på bogstavet u, men de kan anvendes på ethvert bogstav.

Punkter og underpunkter

En væsentlig grund til at skrive en tekst er at man har en forventning om at der er nogen der vil læse den. Derfor bør man opfatte skriveprocessen som en formidlingsproces, og i den sammenhæng er LaTeX et særdeles effektivt formidlingsredskab.

Det sker at tekst-dele består af indviklede forklaringer eller ræsonnementer der med fordel kan stilles op i en række punkter hvoraf nogle måske har underpunkter. I sådanne situationer kan man anvende en eller flere af LaTeXs tre opremsnings-konstruktioner:

`itemize` producerer en række punkter. Eksempel:

EKSEMPEL 2.11

En opremsning:

- Et punkt.
- Et punkt til.
- Endnu et punkt.

Færdig!

```
En opremsning:
\begin{itemize}
  \item Et punkt.
  \item Et punkt til.
  \item Endnu et punkt.
\end{itemize}
Færdig!
```

`enumerate` producerer en række nummererede punkter. Eksempel:

EKSEMPEL 2.12

Nummererede punkter:

1. Et punkt.
2. Et punkt til.
3. Endnu et punkt.

Færdig!

```
Nummererede punkter:
\begin{enumerate}
  \item Et punkt.
  \item Et punkt til.
  \item Endnu et punkt.
\end{enumerate}
Færdig!
```

Bemærk at LaTeX selv sørger for nummereringen.

`description` producerer en række navngivne punkter. Eksempel:

EKSEMPEL 2.13

Husmoderens tip:

Gæs spises Mortens aften, ihukommende Skt. Martin.

Ænder spises til jul, fordi en gås er for stor.

Høns spises i asparges.

God appetit.

```
Husmoderens tip:
\begin{description}
\item[Gæs] spises Mortens aften,
  ihukommende Skt. Martin.
\item[Ænder] spises til jul,
  fordi en gås er for stor.
\item[Høns] spises i asparges.
\end{description}
God appetit.
```

I de tre eksempler er hvert punkt (item) af pladshensyn kun på én linje; hvis et punkt strækker sig over flere linjer, bliver disse automatisk rykket ind så det

navn	niveau
\part	-1
\chapter	0
\section	1
\subsection	2
\subsubsection	3
\paragraph	4
\subparagraph	5

Tabel 2.3 De forskellige 'sectioning commands' og deres nummer i afsnits-hierarkiet. Bemærk at dokumentklasserne artikel og notat ikke kan have kapitler.

er tydeligt at der er tale om en opstilling i punkter. Det kan der findes mange eksempler på i dette hæfte.

Et punkt kan indeholde vilkårlig tekst (og matematiske formler), specielt kan det indeholde en af de tre nævnte opremsningskonstruktioner; på den måde får man underpunkter osv. LaTeX sørger selv for at underpunkter markeres og nummereres på en hensigtsmæssig måde; man kan have op til fire niveauer af opremsninger.

Kapitler og afsnit

En tekst er ofte inddelt i afsnit med overskrifter af forskellige grader, f.eks. *kapitler* der kan være inddelt i *hovedafsnit* der igen kan være inddelt i *mellemrubrikker* (underafsnit). I større afhandlinger kan kapitlerne være samlet i *dele*. I LaTeX markeres sligt ved hjælp af kommandoerne `\part`, `\chapter`, `\section`, `\subsection` osv. der har det pågældende afsnits titel som argument. Eksempel: Indeværende kapitel blev »defineret« ved at skrive

```
\chapter{Enkle tekster uden matematik}
```

og indeværende hovedafsnit ved

```
\section{Kapitler og afsnit}
```

LaTeX tager sig selv af nummereringen af kapitler, afsnit osv; LaTeX véd også at et kapitel i en bog altid begynder på en højreside.

I Tabel 2.3 ses det samlede udvalg af de såkaldte 'sectioning commands' tillige med deres såkaldte niveau. LaTeX nummererer de fire øverste niveauer; hvis man ikke er tilfreds med det, kan man i præamblen (se side 16) ændre tælleren `secnumdepth`; hvis man f.eks. kun vil have nummereret kapitler og afsnit (og dele), altså til og med niveau 1, skal man skrive

```
\setcounter{secnumdepth}{1}
```

Måske skal teksten have et appendiks eller to til sidst. I så fald benyttes kommandoen `\appendix` én gang; alle efterfølgende `\chapter`-kommandoer fremstiller da appendikser (eller måske appendices).

Indholdsfortegnelse

Indholdsfortegnelsen fås ved at skrive kommandoen `\tableofcontents` på det sted i teksten hvor indholdsfortegnelsen ønskes anbragt, og det er jo almindeligvis et sted i begyndelsen.

Første gang LaTeX sætter teksten, bliver indholdsfortegnelsen tom, anden gang har den det rigtige indhold men ofte forkerte sidetal (fordi hvis den selv fylder mere end en eller to sider, så ændres nummereringen af hele den følgende tekst[1]), fra og med tredje gang er den rigtig, med mindre man selv ændrer i teksten.

LaTeX har selv en mening om hvad der skal med i indholdsfortegnelsen, nemlig de fire øverste niveauer af afsnitshierarkiet, jf. Tabel 2.3. Hvis man ikke er tilfreds med det, skal man selv ændre tælleren `tocdepth`; hvis man f.eks. kun vil have niveauerne til og med 1 med, skal man i præamblen (se side 16) skrive

```
\setcounter{tocdepth}{1}
```

Henvisninger

Ofte har man brug for at henvise fra et sted i teksten til et andet – og ofte ønsker læseren sig at forfatteren havde lavet nogle flere henvisninger end tilfældet er. Når man benytter LaTeX, er der ingen undskyldninger for ikke at lave mange og gode henvisninger, for det er uhyre let.

Man kan henvise til *alt* hvad LaTeX selv nummererer (og man bør lade LaTeX nummerere alt hvad der skal nummereres), det være sig kapitler, afsnit, sidenumre, nummererede underpunkter, fodnoter, figurer, tabeller, matematiske formler ... Fremgangsmåden er hver gang den samme: på det sted der skal henvises *til*, placerer man en 'label' (etikette) ved at skrive en kommando af formen `\label{`⟨*etikette*⟩`}`, og på det sted man henviser *fra*, skriver man enten `\ref{`⟨*etikette*⟩`}` for at henvise til den nummererede struktur hvor labelen er anbragt, eller `\pageref{`⟨*etikette*⟩`}` for at henvise til den *side* hvor labelen er anbragt. Eksempel:

EKSEMPEL 2.14

	`\subsection{Numre}\label{nr}`
2.0.1 Numre	`Dette underafsnit har`
Dette underafsnit har nummer 2.0.1 og står på side 14.	`nummer \ref{nr} og står` `på side \pageref{nr}.`

Bemærk at den tekststreng man bruger som ⟨*etikette*⟩, ikke må indeholde de mere specielle bogstaver og tegn som f.eks. æ ø å é â ö ½ §.

Det kan være nyttigt at vide en smule om hvordan LaTeX holder styr på henvisningerne: Når LaTeX går teksten igennem, sker der det at hver gang der *defineres*

[1] Det er grunden til at man i gamle dage ofte nummererede siderne med indledning og indholdsfortegnelse med romertal, uafhængigt af pagineringen af den egentlige tekst: på den måde behøvede sætteren (som dengang var en person og ikke et computerprogram) ikke ompaginere teksten.

en label (med `\label`), så bliver der skrevet oplysninger herom i en hjælpefil[2], og hver gang der optræder en *henvisning* til en label (`\ref` eller `\pageref`), så får henvisningen den værdi som labelen fik i forrige omgang, idet LaTeX nemlig altid begynder med at læse hjælpefilen fra forrige gang. Det betyder at hvis man henviser til en label som ikke eksisterede forrige gang, så kender LaTeX den ikke, og resultatet er en advarsel af formen

```
LaTeX Warning: Reference 'AB.D' on page 6 undefined on input line 126.
```

Man vil også ofte få advarslen

```
LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.
```

Dokumentklasserne

Dokumentklassen bestemmer det *layout* som teksten skal fremstilles efter. Standard LaTeX er udviklet til brug for amerikanere og indeholder blandt andet dokumentklasserne book, report, article og letter samt slides. IMFUFA-LaTeX er beregnet for danskere og engelsk-skrivende europæere og indeholder blandt andet dokumentklasserne bog, rapport, artikel, notat og brev. Dokumentklassernes navne antyder hvad de er beregnet til; nærværende publikation er fremstillet med dokumentklassen bog.

Normalt sættes LaTeX-tekster med en 10 punkt grundskrift, men man kan i dokumentklasseerklæringen vælge en større skriftgrad, se side 19.

Dokumentklassen bestemmer hvilke afsnitstyper der er til rådighed (kapitel, hovedafsnit, mellemrubrik osv., se også Tabel 2.3 på side 13), specielt bemærkes at article/artikel og notat ikke har kapitler, og at letter/brev slet ingen afsnitstyper har.

Forside/titelblad

I de fleste tilfælde ønsker man at det færdige dokument har en forside eller et titelblad eller en side 1 med et særligt udseende; også dette sørger dokumentklassen for hvis man beder om det. Kommandoen `\maketitle` bevirker at der fremstilles en forside eller en første side. Kommandoen skal stå som noget af det første efter `\begin{document}`.

For at LaTeX kan vide hvad forsiden skal indeholde, er man begribeligvis nødt til først at levere forskellige oplysninger, og det skal gøres inden man kalder `\maketitle`, normalt i præamblen (se side 16). Nogle oplysninger er krævede (f.eks. titel), andre er valgfrie idet de på forhånd har fået tildelt en standardværdi (f.eks. har datoen standardværdien `\today`). De forskellige dokumentklasser har forskellige sæt krævede og valgfrie oplysninger.

Dokumentklassen notat kan modtage en *titel*, en *forfatter* (standardværdien er `\today`) og et *sted* (standardværdien er `IMFUFA`); værdierne tildeles med kommandoerne `\title`, `\author` og `\location`. Som eksempel vises i Figur 2.1 en kort tekst af typen notat; heraf fremgår hvordan man tildeler de nævnte størrelser deres værdier.

[2] Hjælpefilens navn bliver ⟨*primærfil*⟩`.aux`.

```
\documentclass{notat}

\author{Morten Vinge\\
        den 12te Januar 1751}
\title{Dagbog}
\location{Thiele}

\begin{document}
\maketitle

En stille, herlig Vinternat!
Alt hvad jeg seer, Blaat eller Hvidt.
Maanen har jaget Stjernerne fra sig,
den vil lyse allene.  Saa deiligt
skinnede den ogsaa engang for mange
-- mange Aar siden, da jeg var Kudsk
for Frøken Sophie.

\end{document}
```

Figur 2.1 Et dokument af type notat.

```
\documentclass{artikel}

\author{Morten
        Vinge\thanks{Landsbydegn}}
\date{den 12te Januar 1751}
\title{Dagbog}

\begin{document}
\maketitle

En stille, herlig Vinternat!
Alt hvad jeg seer, Blaat eller Hvidt.
Maanen har jaget Stjernerne fra sig,
den vil lyse allene.  Saa deiligt
skinnede den ogsaa engang for mange
-- mange Aar siden, da jeg var Kudsk
for Frøken Sophie.

\end{document}
```

Figur 2.2 Et dokument af type artikel.

Dokumentklasserne bog, rapport og artikel forventer en *titel*, en *forfatter* og en *dato*; disse tildeles værdier med kommandoerne `\title`, `\author` og `\date`. Figur 2.2 viser et artikel-eksempel.

Når man angiver værdier for titel, forfatter osv., kan man bruge `\\` for at markere *linjeskift*; der må ikke være tomme linjer, men der må gerne være en linje der kun indeholder et blanktegn (der skrives som `\␣` (»backslash-mellemrum«)).

I tilfælde af *flere forfattere* kan man i bog, rapport og artikel skrive dem med `\and` imellem.

Breve

Dokumentklassen brev er en direkte oversættelse af letter, og der er p.t. ikke gjort noget for at tilpasse den til danske standarder for hvordan breve skal se ud (spørgsmålet er om der overhovedet findes sådanne standarder). LaTeX har helt særlige regler for disse to dokumentklasser. Figur 2.3 viser et brev-skelet.

Dokumentets opbygning; organisering af arbejdet

Et LaTeX-dokument er altid opbygget på følgende måde:

1. Først kommer *dokumentklasseerklæringen* der har formen `\documentclass{`⟨*klasse*⟩`}` eller `\documentclass[`⟨*option-liste*⟩`]{`⟨*klasse*⟩`}` hvor ⟨*option-liste*⟩ er en eller flere options adskilt af komma.

2. Dernæst kommer den såkaldte *præambel*. Det er her man skal angive hvilke *pakker* man ønsker at benytte (læs mere om pakker på side 35); det er også

```
\documentclass{brev}

\name{A. And}
\signature{Anders}
\address{Paradisæblevej 111}

\begin{document}
  \begin{letter}{G. Gearløs \\
                 Newton allé 17 \\
                 Andeby}

    \opening{Kære Gearløs,}

      Som jeg fortalte om i telefonen ...

    \closing{Med håb om hurtig hjælp,}

    \ps{PS. Huset brænder!}

    \cc{Rip\\ Rap\\ Rup}

  \end{letter}
\end{document}
```

Figur 2.3 Skelet til et dokument af typen brev.

her man definerer eventuelle nye kommandoer og udsteder generelle ordrer i det hele taget.

3. Kommandoen `\begin{document}` afslutter præamblen og indleder selve teksten.

   Den første outputproducerende kommando er for det meste `\maketitle`.

4. Den allersidste kommando er altid `\end{document}`.

Hvis man har at gøre med en lang tekst, kan det være upraktisk at have hele teksten stående i én stor fil. Man står sig som regel ved at dele teksten op i mindre filer, f.eks. svarende til kapitler eller afsnit, og så instruere LaTeX om at læse disse i den rigtige rækkefølge. Kommandoen til indlæsning af filer er `\input`; eksempel: `\input{kapitel2.tex}`.[3] Man kan derfor med fordel organisere arbejdet på den måde at man har en overordnet fil, den såkaldte *primærfil*, som indeholder præamblen, og som indlæser de underordnede filer med de enkelte kapitler eller afsnit. Figur 2.4 viser et eksempel på en sådan primærfil.

Undervejs i tilblivelsesprocessen kan man udkommentere en større eller mindre del af input-kommandoerne (ved at sætte % foran), sådan at man f.eks. kan nøjes med at behandle filen `kap3.tex`. Man skal dog være opmærksom på at da LaTeX nummererer kapitler i den rækkefølge de optræder i input-strømmen, så vil det først indlæste kapitel altid få nummeret 1 selv det måske er tænkt som Kapitel 3; det må man affinde sig med – det bliver rigtigt i sidste ende.[4]

[3] Læs nærmere om filnavne på side 95.

[4] Hvis man partout vil give kapitlet et andet nummer, kan man gøre det ved at skrive f.eks. `\setcounter{chapter}{2}` umiddelbart før `\chapter`-kommandoen.

```
\documentclass{bog}
\usepackage{...}

\title{Tredjemodulsprojekt}
\author{A. Nielsen}
\date{November 1999}

\begin{document}
  \maketitle
  \tableofcontents

  \input{forord.tex}
  \input{kap1.tex}
  \input{kap2.tex}
  \input{kap3.tex}
  \input{afrund.tex}
  \input{bilag1.tex}
\end{document}
```

Figur 2.4 Et skelet til en primærfil til et dokument af type bog.

Kapitel 3

Skrifttyper

Normalt vælger LaTeX selv en skrift der passer til omstændighederne, eksempelvis bliver overskrifter automatisk sat i en stor fed skrift. I dette kapitel fortælles lidt om hvordan den almindelige bruger kan ændre forskellige af de parametre der bestemmer skriftsnit og skriftgrad.

Den mest almindelige ændring i skriftbilledet er *fremhævet* tekst; det engelske ord for *fremhæve* er *emphasize*, så derfor hedder kommandoen `\emph`:

Man kan *fremhæve* noget tekst ved at skrive det med en afvigende skrift. Derimod bør man absolut *ikke* sætte en streg under de ord der ønskes fremhævet.	`Man kan \emph{fremhæve} noget tekst ved at skrive det med en afvigende skrift. Derimod bør man absolut \emph{ikke} sætte en streg under de ord der ønskes fremhævet.`

EKSEMPEL 3.1

Valg af skriftgrad

LaTeX vælger selv en skriftgrad (størrelse) der passer til omstændighederne, for eksempel sættes fodnoter med en mindre skrift end grundteksten. Grundteksten sættes normalt i 10 punkt[1]. I dokumentklasseerklæringen har man har dog mulighed for at vælge 11 eller 12 punkt ved at angive `11pt` eller `12pt` som option i dokumentklasseerklæringen, f.eks. sådan: `\documentclass[12pt]{notat}`

Inde i teksten har brugeren mulighed for at vælge skriftgrad med kommandoerne `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge` og `\Huge`. Kommandoernes rækkevidde afgrænses med `{ ... }` eller med `\begin ... \end`. Eksempel:

[1]Eller rettere 10 *point*. – Skriftstørrelser angives ofte i enheden *punkt*. For at det ikke skal være for nemt, er enheden ikke den samme i alle lande og i alle sammenhænge. I Europa bruges enheden *Didôt punkt* (dd), 1 dd = 0.376 mm = $^1/_{72}$ fransk tomme. I Amerika og England bruger man enheden *American point* (pt), 1 pt = 0.351 mm = $^1/_{72.27}$ inch. (Man kan læse mere om det typografiske punkt i Reimer (1990).)

I PostScript-sammenhæng opererer man med enheden Big point (bp), 1 bp = 0.351 mm = $^1/_{72}$ inch.

EKSEMPEL 3.2

Et lille bitte nøk.
Et lille bitte nøk.
	`Et {\tiny lille bitte} nøk.`
	
	`Et \begin{tiny}lille bitte\end{tiny} nøk.`

Hvis man benytter disse kommandoer, skal man være opmærksom på at linjeafstanden afhænger af den valgte skriftgrad, og på at LaTeX sætter et helt afsnit ad gangen med den linjeafstand der er gældende på det sted hvor afsnittet er færdigt. Derfor skal man som regel sætte et »afsnit slut«-mærke *inden* man afslutter den forstørrede eller formindskede tekst; »afsnit slut«-mærket er som altid en blank linje eller kommandoen `\par` (`\bpar` kan også bruges).

EKSEMPEL 3.3

Her vises hvordan man *ikke* skal gøre. Afsnittet med lille skrift bliver sat med normalskriftens linjeafstand.
Her er normal skriftstørrelse.
	`{\tiny Her vises hvordan man \emph{ikke} skal gøre. Afsnittet med lille skrift bliver sat med normalskriftens linjeafstand.}`
	
	`Her er normal skriftstørrelse.`

EKSEMPEL 3.4

Her afsluttes afsnittet med lille skrift med et »nyt afsnit«-mærke, så derfor bliver afsnittet sat med den rigtige linjeafstand.
Her er normal skriftstørrelse.
	`{\tiny Her afsluttes afsnittet med lille skrift med et \textqm{nyt afsnit}-mærke, så derfor bliver afsnittet sat med den rigtige linjeafstand.\par}`
	
	`Her er normal skriftstørrelse.`

Valg af skriftsnit

Tekstens brødskrift, dvs. den skrift som den altovervejende del af dokumentet sættes med, bør være en neutral og letlæselig skrift, normalt fra den skriftfamilie der kaldes **antikva** (engelsk: Roman). LaTeX benytter automatisk den antikvaskrift der ses som brødskrift i nærværende hæfte, og som hedder European Computer Modern Roman.

For at fremhæve enkelte tekstdele kan man tage særlige virkemidler i brug, som for eksempel at vælge en ***hældende antikva****-skrift. En anden mulighed er en* ***kursiv****-skrift. Kursiv-skriftens bogstaver og tegn adskiller sig væsentligt fra antikva-skriftens, se f.eks. på bogstaver som a f g æ.* MAN KAN OGSÅ BENYTTE **KAPITÆLSKRIFT**; DEN ER KARAKTERISTISK VED AT DE SMÅ BOGSTAVER ER MAGEN TIL DE STORE, BLOT MINDRE. `Til særlige formål kan skrivemaskineskrift være velegnet.` En helt anden skriftfamilie er **blokskrift** (grotesk), der ofte kaldes sans serif fordi blokskrift-bogstaverne (modsat antikva-bogstaverne) mangler serifferne, dvs. de små ekstra streger der er hvor bogstaverne ender. Hvis man uafladeligt skifter skriftsnit, opnår man at teksten bliver temmelig ulæselig.

LaTeX har et meget elegant system til skift mellem de forskellige skriftformer. Ideen er at en skrift kan beskrives ved fire attributter hvis betegnelser (på engelsk) er ⟨*family*⟩, ⟨*series*⟩, ⟨*shape*⟩ og ⟨*size*⟩, og som kan vælges uafhængigt af hverandre. Attributterne kan have følgende værdier:

⟨*family*⟩ kan være

- `rm` Antikva (engelsk: Roman). Dette er standard-værdien.
- `sf` Blokskrift (eller Sans serif)
- `tt` `Skrivemaskineskrift` (engelsk: Typewriter)

⟨*series*⟩ kan være

- `md` Medium. Dette er standard-værdien.
- `bf` **Fed** (engelsk: Boldface)

⟨*shape*⟩ kan være

- `up` Opret (engelsk: Upright). Dette er standard-værdien.
- `it` *Kursiv* (engelsk: Italic)
- `sl` *Hældende* (engelsk: Slanted)
- `sc` KAPITÆLER (engelsk: Small Caps)

⟨*size*⟩ kan være en af de 10 værdier der nævnt på side 19.

Man kan vælge ⟨*family*⟩, ⟨*series*⟩ og ⟨*shape*⟩ på to måder, enten med en *kommando* eller med en *erklæring*. Benytter man en kommando, er syntaksen ⟨*kommando*⟩{⟨*tekst*⟩}; eksempel:

Dette er **fed** skrift, og dette er *kursiv*.	`Dette er \textbf{fed} skrift,` `og dette er \textit{kursiv}.`

EKSEMPEL 3.5

Benytter man en erklæring, kan man vælge mellem en konstruktion af formen `\begin{`⟨*navn*⟩`}` ⟨*tekst*⟩ `\end{`⟨*navn*⟩`}` og en konstruktion af formen `{\`⟨*navn*⟩ ⟨*tekst*⟩`}`; eksempel:

Her er fed skrift.
Her er kursiv-skrift.
	`\begin{bfseries}`
	`  Her er fed skrift.`
	`\end{bfseries}\par`
	`{\itshape Her er kursiv-skrift.}`

EKSEMPEL 3.6

I Tabel 3.1 er vist de forskellige kommandoer og erklæringer til skriftvalg.

Til demonstration af at man kan variere de forskellige skriftattributter uafhængigt af hverandre, begynder vi med følgende eksempel:

Han skulle spille **Hamlets faders *Ånd*!**	`Han skulle spille {\bfseries` `Hamlets faders \emph{Ånd}!}`

EKSEMPEL 3.7

Nu kan vi ændre det hele til blokskrift med et enkelt indgreb:

EKSEMPEL 3.8

Han skulle spille **Hamlets faders *Ånd*!**	`\sffamily`
	`Han skulle spille {\bfseries`
	`Hamlets faders \emph{Ånd}!}`

Enhver af de $3 \times 2 \times 4 = 24$ kombinationer af ⟨*family*⟩, ⟨*series*⟩ og ⟨*shape*⟩ er tilladt i hver af de 10 størrelser, men dermed er ikke sagt at alle 240 kombinationer findes i en given LaTeX-installation; nogle skrifter hører ikke med i standardopsætningen af LaTeX, og nogle skrifter eksisterer simpelthen ikke – der er f.eks. ingen kursiv blokskrift. Heldigvis er LaTeX indrettet sådan at når man beder om en ikke-eksisterende skrift, så får man automatisk en erstatning – f.eks. bliver kursiv blokskrift automatisk erstattet af hældende blokskrift.[2] I Tabel 3.2 er vist hvad der »i skrivende stund« (18. maj 1999) faktisk kommer ud af at forsøge de 24 kombinationer i 10 punkt når man bruger IMFUFAs LaTeX-installation på MMF.

»Gammeldags« tal

De så moderne »gammeldags« tal, eller rettere mediævaltal eller *hængende tal*, (som i 3.14159) fås med kommandoen `\oldstylenums`; den udgave af kommandoen som findes i IMFUFA-LaTeX respekterer de forskellige skriftvalgskommandoer:

EKSEMPEL 3.9

J.S.Bach levede fra 1685 til 1750.	`J.S.Bach levede fra`
	`\oldstylenums{1685} til`
	`\oldstylenums{1750}.`

J.S.Bach levede fra 1685 til 1750.	`\bfseries\slshape`
	`J.S.Bach levede fra`
	`\oldstylenums{1685} til`
	`\oldstylenums{1750}.`

[2] Hvis erstatningen ikke er særlig god, kommer der en 'Warning' i `log`-filen, f.eks.
```
LaTeX Font Warning: Font shape 'T1/cmss/m/sc' in size <10> not available
(Font) Font shape 'T1/cmr/m/sc' tried instead on input line 291.
```
hvor beskeden er, at der ikke findes nogen kapitælskrift (`sc`) i Medium (`m`) i blokskriften (`cmss`), og at der i stedet er benyttet kapitælskrift i Medium i antikvaskriften (`cmr`).

kommando	erklæring	beskrivelse
`\textrm{...}`	`{\rmfamily ...}`	Antikva
`\textsf{...}`	`{\sffamily ...}`	Blokskrift
`\texttt{...}`	`{\ttfamily ...}`	Skrivemaskine
`\textmd{...}`	`{\mdseries ...}`	Medium
`\textbf{...}`	`{\bfseries ...}`	**Fed**
`\textup{...}`	`{\upshape ...}`	Opret
`\textit{...}`	`{\itshape ...}`	*Kursiv*
`\textsl{...}`	`{\slshape ...}`	*Hældende*
`\textsc{...}`	`{\scshape ...}`	KAPITÆLER
`\emph{...}`	`{\em ...}`	*Fremhævet*
`\textnormal{...}`	`{\normalfont ...}`	Brødskriften

Tabel 3.1 Skriftvalg: kommandoer og erklæringer

	rm	sf	tt
up	Sola scriptura **Sola scriptura**	Sola scriptura **Sola scriptura**	Sola scriptura Sola scriptura
it	*Sola scriptura* ***Sola scriptura***	*Sola scriptura* ***Sola scriptura***	*Sola scriptura* *Sola scriptura*
sl	*Sola scriptura* ***Sola scriptura***	*Sola scriptura* ***Sola scriptura***	Sola scriptura Sola scriptura
sc	SOLA SCRIPTURA **SOLA SCRIPTURA**	SOLA SCRIPTURA **Sola scriptura**	SOLA SCRIPTURA Sola scriptura

Tabel 3.2 Her ses resultaterne af alle 24 kombinationer af ⟨*family*⟩, ⟨*series*⟩ og ⟨*shape*⟩.
Rækkerne svarer til de forskellige værdier af ⟨*shape*⟩ og søjlerne til de forskellige værdier af ⟨*family*⟩; i den enkelte celle svarer øverste linje til ⟨*series*⟩=Medium og nederste linje til ⟨*series*⟩=Bold.
Når tabellen ikke indeholder 24 *forskellige* skrifteksempler, skyldes det at en del af skrifttyperne ikke findes i den aktuelle LaTeX-installation.

Concrete	Quod scripsi, scripsi
ss demibold condensed	**Quod scripsi, scripsi**
Times	Quod scripsi, scripsi
Palatino	Quod scripsi, scripsi
NewCenturySchoolbook	Quod scripsi, scripsi
Bookman	Quod scripsi, scripsi
Chancery	*Quod scripsi, scripsi*
AvantGarde	Quod scripsi, scripsi
Helvetica	Quod scripsi, scripsi

Tabel 3.3 Prøver på andre skrifter

Andre skrifter

Det kan jo ske at man trænger til at se nogle andre skrifter end Computer Modern, men det er der også råd for.

Et brugbart alternativ er skriften Concrete som Donald Knuth designede til brug i en bog med titlen *Concrete Mathematics*[3]. Man kan vælge Concrete-skriften ved hjælp af Standard LaTeX pakken ccfonts (Schmidt, 1999); denne pakke ændrer i øvrigt matematikskriften tilsvarende.

Hvis man bruger IMFUFA-LaTeX, kan man med fordel vælge Concrete-skriften ved at angive `ccfonts` som en option i dokumentklasseerklæringen, f.eks. sådan: `\documentclass[ccfonts]{bog}`. Derved sker der automatisk nogle ekstra småjusteringer som følge af at der ikke findes nogen fed udgave af Concrete; IMFUFA-LaTeX bruger så i de fleste tilfælde »sans serif demibold condensed« (`T1/cmss/sbc/n`).

En anden mulighed er at benytte PostScriptskrifter. De fleste PostScriptprintere skulle have skrifter som Times, Palatino, NewCenturySchoolbook, Bookman, Chancery, AvantGarde og Helvetica, og man kan få sit dokument sat med en af disse ved at vælge den relevante af pakkerne times[4], palatino[5], newcent[6], bookman[7], chancery[8], avant[9] og helvet[10]. Bemærk at disse pakker kun ændrer de skrifter der bruges til *tekst*. Matematikskriften er den sædvanlige, og den ser ikke godt ud sammen med PostScriptskrifterne. Hvis man bruger Times, kan man dog inkludere pakken mathptm (foruden times), derved bliver de fleste matematiske symboler sat med en Times-skrift. Bemærk også at ikke alle viewere kan vise PostScriptskrifter.

[3] Konkret matematik er – bl.a. – *kon*tinuert og dis*kret* matematik til brug i datalogi.
[4] vælger Times som `\rmfamily` og Helvetica som `\sffamily`.
[5] vælger Palatino som `\rmfamily` og Helvetica som `\sffamily`.
[6] vælger NewCenturySchoolbook som `\rmfamily` og AvantGarde som `\sffamily`.
[7] vælger Bookman som `\rmfamily` og AvantGarde som `\sffamily`.
[8] vælger Chancery som kursivskriften i `\rmfamily`.
[9] vælger AvantGarde som `\sffamily`.
[10] vælger Helvetica som `\sffamily`.

Kapitel 4

For let øvede

Skemaer; tabular-konstruktioner

Undertiden har man brug for at stille ting op i skematiske oversigter sådan som der ses eksempler på i tabellerne på side 13, 23 og 23. Skemaer kan fremstilles ved hjælp af `tabular`-konstruktioner. Det vil vi nu demonstrere ved hjælp af en række eksempler der i øvrigt viser udvalgte valutakurser den 11. november 1996. Her er første udgave af eksemplet:

D-mark	384.62
Pund	953.66
Franske fr.	113.71
Gylden	342.98
Svenske kr.	87.59
Norske kr.	91.58
Islandske kr.	8.76

```
\begin{tabular}{lr}
  D-mark        & 384.62 \\
  Pund          & 953.66 \\
  Franske fr.   & 113.71 \\
  Gylden        & 342.98 \\
  Svenske kr.   &  87.59 \\
  Norske kr.    &  91.58 \\
  Islandske kr.&    8.76
\end{tabular}
```

EKSEMPEL 4.1

Hele `tabular`-konstruktionen omgives af `\begin{tabular}` `\end{tabular}`. Umiddelbart efter `\begin{tabular}` skal man angive hvor mange søjler der skal være, og om de skal venstrejusteres, centreres eller højrejusteres; det gøres ved at man for hver ønsket søjle skriver ét af bogstaverne `l` (left), `c` (centre) og `r` (right). I eksemplet ønskes der altså to søjler hvoraf den første skal venstrejusteres og den anden højrejusteres. Skemaets indhold skrives række for række; tegnet `&` er skilletegn mellem de enkelte felter, og tegnet `\\` er skilletegn mellem de enkelte rækker. – I eksemplet har brugeren skrevet input-teksten pænt op med alle `&`'erne under hinanden; dette er udelukkende af æstetiske grunde og fordi når man skriver tingene pænt og systematisk op, er det lettere at opdage eventuelle fejl, LaTeX er fuldstændig ligeglad.

Man kan få lodrette streger mellem søjlerne ved at putte en `|` ind de ønskede steder i listen af søjle-specifikatorer, og man kan få vandrette streger med kommandoen `\hline`; bemærk at der skal være `\\` efter den linje der skal understreges:

EKSEMPEL 4.2

D-mark	384.62
Pund	953.66
Franske fr.	113.71
Gylden	342.98
Svenske kr.	87.59
Norske kr.	91.58
Islandske kr.	8.76

```
\begin{tabular}{|l|r||}
  \hline
  D-mark        & 384.62 \\
  Pund          & 953.66 \\
  Franske fr.   & 113.71 \\
  Gylden        & 342.98 \\
  \hline
  Svenske kr.   &  87.59 \\
  Norske kr.    &  91.58 \\
  Islandske kr.&   8.76 \\
  \hline\hline
\end{tabular}
```

Nu vil vi gerne have skrevet teksten 11.11.96 ind som en slags overskrift, og samtidig fjerner vi de fleste af stregerne igen. Her er første forsøg:

EKSEMPEL 4.3

11.11.96	
D-mark	384.62
Pund	953.66
Franske fr.	113.71
Gylden	342.98
Svenske kr.	87.59
Norske kr.	91.58
Islandske kr.	8.76

```
\begin{tabular}{lr}
  11.11.96 \\
  \hline
  D-mark        & 384.62 \\
  Pund          & 953.66 \\
  Franske fr.   & 113.71 \\
  Gylden        & 342.98 \\
  Svenske kr.   &  87.59 \\
  Norske kr.    &  91.58 \\
  Islandske kr.&   8.76
\end{tabular}
```

Overskriften bliver venstrejusteret, og det var ikke meningen. En udvej er at benytte kommandoen `\multicolumn` der har tre argumenter: antal søjler den skal spænde over (her 2), en søjle-specifikator (her `c`) og selve indholdet (her `11.11.96`):

EKSEMPEL 4.4

11.11.96	
D-mark	384.62
Pund	953.66
Franske fr.	113.71
Gylden	342.98
Svenske kr.	87.59
Norske kr.	91.58
Islandske kr.	8.76

```
\begin{tabular}{lr}
  \multicolumn{2}{c}{11.11.96} \\
  \hline
  D-mark        & 384.62 \\
  Pund          & 953.66 \\
  Franske fr.   & 113.71 \\
  Gylden        & 342.98 \\
  Svenske kr.   &  87.59 \\
  Norske kr.    &  91.58 \\
  Islandske kr.&   8.76
\end{tabular}
```

Nu vil vi prøve at sætte overskrifter på hver enkelt søjle. Her er et forsøg:

Valuta	Kurs
D-mark	384.62
Pund	953.66
Franske fr.	113.71
Gylden	342.98
Svenske kr.	87.59
Norske kr.	91.58
Islandske kr.	8.76

```
\begin{tabular}{lr}
  Valuta & Kurs \\
  \hline
  D-mark        & 384.62 \\
  Pund          & 953.66 \\
  Franske fr.   & 113.71 \\
  Gylden        & 342.98 \\
  Svenske kr.   &  87.59 \\
  Norske kr.    &  91.58 \\
  Islandske kr.&   8.76
\end{tabular}
```

EKSEMPEL 4.5

Resultatet er ikke det bedst tænkelige, det ville måske se bedre ud hvis overskrifterne Valuta og Kurs stod centreret over de respektive søjler. Det gøres på følgende måde; vi tager udgangspunkt i et skema hvor overskrifterne står pænt centreret over hver sin søjle:

Valuta	Kurs
A	B

```
\begin{tabular}{cc}
  Valuta & Kurs \\
  \hline
  A & B
\end{tabular}
```

EKSEMPEL 4.6

Det er sådan at indholdet i et enkelt felt i et skema godt kan være et skema. Derfor kan vi erstatte `A` med et skema indeholdende valuta-navnene og `B` med et skema indeholdende kurserne:

Valuta	Kurs
D-mark	384.62
Pund	953.66
Franske fr.	113.71
Gylden	342.98
Svenske kr.	87.59
Norske kr.	91.58
Islandske kr.	8.76

```
\begin{tabular}{cc}
  Valuta & Kurs \\
  \hline
  \begin{tabular}{l}
    D-mark\\ Pund\\ Franske fr.\\
    Gylden\\ Svenske kr.\\
    Norske kr.\\ Islandske kr.
  \end{tabular}
  &
  \begin{tabular}{r}
    384.62\\ 953.66\\ 113.71\\
    342.98\\ 87.59\\ 91.58\\ 8.76
  \end{tabular}
\end{tabular}
```

EKSEMPEL 4.7

Skemaets placering

En ting er at få skemaets indhold rigtigt, noget andet er at få placeret skemaet rigtigt i forhold til den omgivende tekst. Hvordan det kan gøres, viser vi i det følgende hvor vi nu benytter et andet og mindre skema som eksempel.

LaTeX behandler et skema som et tekstobjekt der indgår i den almindelige tegnstrøm:

EKSEMPEL 4.8

LaTeX betragter skemaet	A B C D	som
et almindeligt tekstobjekt.		

```
\LaTeX{} betragter skemaet
\begin{tabular}{cc}
  A & B\\ C & D
\end{tabular}
som et almindeligt tekstobjekt.
```

Som det ses bliver skemaet centreret horisontalt i forhold til teksten, men det kan man let ændre ved at angive plads-specifikatoren `b` (bund) eller `t` (top):

EKSEMPEL 4.9

LaTeX betragter skemaet	A B C D	som
et almindeligt tekstobjekt.		

```
\LaTeX{} betragter skemaet
\begin{tabular}[t]{cc}
  A & B\\ C & D
\end{tabular}
som et almindeligt tekstobjekt.
```

Hvis man vil have skemaet til at stå fremhævet (og centreret) midt på siden, opnås det med mindst skrivearbejde ved at sætte `\[ \]` rundt om:

EKSEMPEL 4.10

Her bliver skemaet fremhævet (og centreret):

A B
C D

Det kom altså til at se sådan ud.

```
Her bliver skemaet fremhævet
(og centreret):
\[\begin{tabular}{cc}
  A & B\\ C & D
\end{tabular}\]
Det kom altså til at se sådan ud.
```

(Kommandoerne `\[ \]` laver »fremhævet matematik«, se side 40, og man kan derfor ikke skrive almindelig tekst imellem dem.)

En side på siden: `minipage`-konstruktioner

En *minipage* specificeres efter opskriften

`\begin{minipage}`⟨*bredde*⟩ ⟨*indhold*⟩ `\end{minipage}`

Der sker da det at ⟨*indhold*⟩ bliver sat på en mini-side af den angivne bredde. Samtidig opfatter LaTeX en minipage som ét enkelt tekstelement i forhold til den større side.

Den bredde (width) den skal angives, skal være en længde (i LaTeX-forstand), for eksempel `36mm` eller `0.7\textwidth` eller `0.5\linewidth`.

En minipage kan justeres horisontalt i forhold til den omgivende tekst med specifikatorerne `c` (centreret), `t` (top) og `b` (bund), ligesom for skemaers vedkommende (Eksempel 4.9); man skriver f.eks. `\begin{minipage}[t]{35mm}`.

Her er et citat fra Robert Musils *Manden uden egenskaber*:

> Egentlig var det kun matematikerne selv og deres disciple, naturforskerne, der levede i uvidenhed om disse farer, de fornemmede lige så lidt af alt dette i deres sjæl som cykelryttere, der flittigt tramper løs og ikke lægger mærke til andet i verden end den forankørendes baghjul. Om Ulrich kunne man derimod med sikkerhed sige det ene, at han elskede matematikken på grund af de mennesker, der ikke kunne udstå den. Han var mindre videnskabeligt end menneskeligt forelsket i videnskaben. Han så, at den i alle spørgsmål, hvor den betragter sig som kompetent, tænker anderledes end almindelige mennesker. Hvis man i stedet for videnskabelige anskuelser satte livsanskuelse, i stedet for hypotese forsøg og i stedet for sandhed handling, var der ingen betydelig naturforsker eller matematiker, hvis livsværk ikke langt overgik historiens største bedrifter.

```
Her er et citat fra Robert Musils
\textsl{Manden uden egenskaber}:
\[\begin{minipage}{0.85\linewidth}
\scriptsize
Egentlig var det kun matematikerne
selv og deres disciple,
naturforskerne, der levede i
uvidenhed om disse farer

...

hvis livsværk ikke langt overgik
historiens største bedrifter.
\end{minipage}\]
```

EKSEMPEL 4.11

Sidens udformning og opbygning

LaTeX sørger automatisk for en passende udformning af siden, herunder for sådanne ting som størrelse af marginer, nummerering af sider og placering af sidetal. I visse dokumentklasser har siderne desuden en såkaldt *levende kolumnetitel* (således som det ses øverst på siderne i dette hæfte). LaTeX benytter særlige regler for udformningen af den første side i et kapitel og den første side i det samlede dokument.

Brugeren kan vælge mellem forskellige standardudformninger af den normale side ved at give en kommando af formen `\pagestyle{`⟨*stil*⟩`}`, hvor ⟨*stil*⟩ kan være `empty`, `plain`, `headings` eller `lheadings` (sidstnævnte dog kun i dokumentklasserne bog og rapport). En `\pagestyle`-kommando vil normalt stå i præamblen fordi man normalt vil bruge samme stil i hele dokumentet. En sjælden gang kan man have brug for at ændre stilen for en enkelt side; det gøres med kommandoen `\thispagestyle`.

Hvis man som ⟨*stil*⟩ vælger `headings` eller `lheadings`, får siderne en levende kolumnetitel med information om kapitel og afsnit. Det foregår (af sig selv) på den måde at LaTeX henter information fra `\chapter`- og `\section`-kommandoerne og benytter den til kolumnetitlerne. Brugeren har mulighed for at gribe ind på to måder:

1. Hvis et kapitel har en meget lang titel, kan man angive en kortere titel som et ekstra argument til `\chapter`, dvs. man skriver noget i retning af

   `\chapter[Kort titel]{Lang titel}`

   hvorved »Kort titel« bliver brugt i kolumnetitlen (og i indholdsfortegnelsen). – Der gælder noget tilsvarende for `\section`.

2. Man kan benytte kommandoer af formen

   \markboth{⟨*venstretitel*⟩}{⟨*højretitel*⟩} og \markright{⟨*højretitel*⟩}

   for at ændre de titler der er hentet fra de senest foregående kapitel- og afsnitskommandoer. For at have virkning på en given side skal kommandoen skrives *inden* LaTeX afslutter fremstillingen af den pågældende side.

   Eksempelvis kan man skrive \markboth{}{} umiddelbart foran en \chapter-kommando for at undertrykke kolumnetitlen på den modstående side til den side hvor det nye kapitel begynder.

Sideskift

LaTeX betjener sig af en del indbyggede regler for hvornår teksten må og hvornår den skal begynde på en ny side, men det er ikke altid at resultatet bliver tilfredsstillende. Så må brugeren gribe ind. Der er kommandoer der på forskellig måde fremprovokerer eller forhindrer et sideskift:

- Kommandoen \newpage laver råt og brutalt et sideskift.
- Kommandoen \clearpage sørger for at eventuelle ophobede figurer og tabeller mv. bliver skrevet ud (jf. side 34), og derefter begyndes på ny side.
- Kommandoen \cleardoublepage virker ligesom \clearpage, blot begyndes der på en ny højreside.
- I nødsfald kan man forstørre (eller formindske) en side med en given længde ved hjælp af kommandoen \enlargethispage. Hvis man for eksempel skriver

  \enlargethispage{2\baselineskip}

  så bliver siden gjort to linjer længere (størrelsen \baselineskip er afstanden mellem underkanten af på hinanden følgende tekstlinjer).

Undertiden kan man få brug for sider uden tekst, men som i øvrigt har samme udformning som de øvrige sider (f.eks. har sidetal). Til det formål har IMFUFA-LaTeX kommandoen \emptypages der som argument har det ønskede antal tomme sider. Eksempel: \emptypages{5}.

Linjeskift

Normalt klarer LaTeX selv at skifte linje på de rigtige steder, men en sjælden gang må man gribe ind med »håndkraft«, for eksempel hvis linjen indeholder lange udelelige ord. Følgende kommandoer kan være nyttige at kende:

- Kommandoen »blank linje« (eller kommandoen \par) betyder *begynd på nyt afsnit*; alt efter omstændighederne viser dette sig enten ved at der begyndes på ny linje som indrykkes, eller ved at der begyndes på ny linje som ikke indrykkes, men som har ekstra afstand til den foregående tekst.

- Kommandoen `\bpar` betyder *begynd på et nyt afsnit, men lav lidt ekstra afstand* (se Eksempel 2.5 på side 8). Denne kommando findes kun i IMFUFA-LaTeX.
- Kommandoen `\\` (eller `\newline`) betyder *begynd på ny linje.*
- Kommandoen `\indent` laver *indrykning.* Kommandoen `\noindent` ophæver en eventuel indrykning.
- For at forhindre at LaTeX deler et bestemt ord, eller for at forhindre meget store ordmellemrum, kan man putte den »besværlige« tekst ind i en `\mbox`.

(Læs også om orddeling, side 37.)

Centreret tekst, citater mm.

Man kan have brug for at skrive særlige tekstdele på en måde så de klart adskiller sig fra den omgivende tekst. Der er flere muligheder.

Centreret tekst

Som vi så i Eksempel 4.10, kan man centrere et skema ved at sætte det ind mellem `\[` og `\]`. Man kunne i stedet have anbragt skemaet i en `center`-konstruktion; denne kan endvidere benyttes til at skrive centrerede tekstlinjer.

Her er hovedteksten.
 	
Her er noget
tekst
der er centreret.
 	
Her fortsætter den normal tekst.
	`Her er hovedteksten.`
	`\begin{center}`
	`  Her er noget\\ tekst\\`
	`  der er centreret.`
	`\end{center}`
	`Her fortsætter den normal tekst.`

EKSEMPEL 4.12

Som det ses, kommer der ekstra linjeafstand ind mellem normalteksten og den centrerede tekst. Hvis man ikke ønsker det, må man betjene sig af andre kneb:

Her er en ganske almindelig tekstlinje.
centreret tekst
venstrejusteret tekst
højrejusteret tekst
Her fortsætter den normale tekst.
	`Her er en ganske almindelig`
	`tekstlinje.\\`
	`\centerline{centreret tekst}\\`
	`\leftline{venstrejusteret tekst}\\`
	`\rightline{højrejusteret tekst}\\`
	`Her fortsætter den normale tekst.`

EKSEMPEL 4.13

Citater

Tekstdele så som længere citater og lignende kan skrives i en `quote`- eller `quotation`-konstruktion. I eksemplerne sætter vi desuden teksten i en mindre grad.

EKSEMPEL 4.14

quote:

Højre- og venstremargin rykkes ind i forhold til den omgivende tekst.

Bemærk hvordan »nyt afsnit« vises.

```
\texttt{quote}:
\begin{quote}\scriptsize
 Højre- og venstremargin rykkes
 ind i forhold til den omgivende
 tekst.

 Bemærk hvordan \textqm{nyt
 afsnit} vises.
\end{quote}
```

EKSEMPEL 4.15

quotation:

Højre- og venstremargin rykkes ind i forhold til den omgivende tekst.

Bemærk hvordan »nyt afsnit« vises.

```
\texttt{quotation}:
\begin{quotation}\scriptsize
 Højre- og venstremargin rykkes
 ind i forhold til den omgivende
 tekst.

 Bemærk hvordan \textqm{nyt
 afsnit} vises.
\end{quotation}
```

Vers

LaTeX har en verse-konstruktion der efter navnet at dømme er tiltænkt poesi:

EKSEMPEL 4.16

1. Jeg kan se på dine øjne
   at du har en anden kær;
   min sødeste veninde,
   sig mig dog, hvem det er.
2. Dine sorte-brune øjne
   og din røde rosenmund,
   de har lagt oppå mit hjerte
   en byrde så tung.

```
\begin{verse}
 \item[1.]
 Jeg kan se på dine øjne \\
 at du har en anden kær; \\
 min sødeste veninde, \\
 sig mig dog, hvem det er.

 \item[2.]
 Dine sorte-brune øjne \\
 og din røde rosenmund, \\
 de har lagt oppå mit hjerte \\
 en byrde så tung.
\end{verse}
```

Figurer og tabeller

Almindeligvis bytter LaTeX ikke om på rækkefølgen af ord og formler osv., alt kommer i samme rækkefølge som forfatteren skriver det. Figurer og tabeller er imidlertid en undtagelse; de bliver af pladsmæssige og æstetiske grunde opfattet som selvstændige tekstenheder der blot skal placeres et eller andet passende sted i nærheden af den sammenhæng hvori de indgår. Tekstobjekter som figurer og tabeller kaldes i LaTeX-jargonen for *floats* fordi deres placering er lidt flydende.

En liden demonstrationsfigur

Figur 4.1 Figurtekst.

LaTeX har et sæt interne regler for hvordan floats skal placeres (helst øverst på en side, eventuelt nederst; kun en vis brøkdel af siden må optages af floats, medmindre det er en side der kun bruges til floats, osv.). Det skal understreges at hele pointen med floats netop er at deres placering *ikke* er fuldstændig fast, og at de altså *ikke* er tænkt til at skulle stå et helt bestemt sted i teksten.

I det følgende omtales hvordan man fremstiller figur-objekter; tabel-objekter fremstilles på tilsvarende måde, man skal blot erstatte `figure` med `table`.

Figurer

Figurer fremstilles efter grundopskriften

```
\begin{figure}
  ⟨figurindhold⟩
 \caption{\label{⟨etikette⟩} ⟨figurtekst⟩}
\end{figure}
```

Som ⟨*figurindhold*⟩ kan man skrive vilkårlige LaTeX-kommandoer, f.eks.

- almindelig tekst, der med fordel kan skrives i en `minipage`, se side 28;
- en `\vspace`-kommando der afsætter en vis mængde lodret (vertikal) plads (f.eks. `\vspace{3cm}`) hvor man senere kan klistre et billede ind;[1]
- en kommando der indlæser et billede fremstillet uden for TeX, for eksempel i eps-format, se side 69;
- en `picture`-konstruktion, se side 71;
- et eller flere skemaer, se side 25.

Bemærk at en eventuel *label* der skal bruges ved henvisning til figuren, skal stå i forbindelse med ⟨*figurtekst*⟩.

Eksempel: De følgende linjer frembringer Figur 4.1 som LaTeX vil placere et sted her i nærheden (desværre tillader LaTeX ikke figurer og tabeller inde i de sædvanlige eksempel-kasser):

```
\begin{figure}
  {\Huge\bfseries En liden demonstrationsfigur}
  \caption{\label{fig1}Figurtekst.}
\end{figure}
```

[1] Det er naturligvis højst usportsligt at klistre figurindholdet ind, man burde lave det hele på computeren (og helst i LaTeX), også selv om det forlænger arbejdstiden med en uge!

Tips og tricks

LaTeX kan somme tider have vanskeligheder med at placere floats, især hvis de fylder meget i højden. Hvis LaTeX ikke kan finde ud af at placere en figur eller tabel, så bliver den gemt til sidst (eller til der kommer en `\clearpage`-kommando); og eftersom der aldrig byttes om på den indbyrdes rækkefølge af f.eks. figurer, så betyder det at hvis én figur gemmes til sidst, så gemmes også alle de følgende figurer til sidst. Her er forskellige tips og fiduser der kan afhjælpe disse og andre problemer i forbindelse med figurer og tabeller:

1. Sørg for at der er en blank linje umiddelbart før `\begin{figure}` og en blank linje umiddelbart efter `\end{figure}`.

2. Man kan tilføje `[tb]` efter `\begin{figure}` for at fortælle at figuren kan placeres øverst (top) eller nederst (bund) på siden; det véd LaTeX ganske vist i forvejen, men erfaringsmæssigt er det (også her) godt at repetere.

3. Hvis LaTeX stadig har problemer med at placere figuren det rigtige sted, kan det hjælpe at erstatte `[tb]` med `[!tb]`; udråbstegnet får LaTeX til at forvalte de æstetiske regler lidt lempeligere.

4. I stedet for `[tb]` kan man skrive `[p]` (`p` som page (side)). Så bliver figuren placeret på en side der kun bruges til floats.

5. Som ⟨*figurtekst*⟩ kan man skrive nogenlunde hvad man har lyst til; men hvis figurteksten er meget lang eller indeholder blanke linjer, så kommer der en overraskende fejlmeddelelse, nemlig noget i retning af

   ```
   Runaway argument?
   {\protect \numberline {\csname the\@captype \endcsname }{\ignorespaces \ETC.
   ! Paragraph ended before \addcontentsline was complete.
   <to be read again>
                       \par
   ```

   Problemet er at LaTeX har en intern liste med alle figurtekster, og de enkelte elementer i denne liste kan kun have en begrænset længde. Listen benyttes hvis brugeren beder om en »Liste over figurer« (med kommandoen `\listoffigures`).

   Udvejen er at bede LaTeX om at skrive en kortere tekst i sin interne liste; det gøres ved at skrive den korte tekst som et valgfrit argument til `\caption`, for eksempel `\caption[kort tekst]{lang tekst}`. Hvis man ikke har tænkt sig at bede om en »Liste over figurer«, kan man bare skrive et eller andet vilkårligt bogstav som kort tekst, f.eks. `\caption[k]{lang tekst}`.

Dobbeltfigurer

Hvis man har meget små figurer/tabeller, kan det måske være formålstjenligt at placere dem ved siden af hinanden sådan som det ses i Figur 4.2 og Figur 4.3 der er fremstillet på følgende måde:

Figur 1

Figur 4.2 Dette er den første figur.

Figur 2

Figur 4.3 Dette er den anden figur.

```
\begin{figure}
  \hfill
  \begin{minipage}{0.3\textwidth}
    {\Huge\bfseries Figur 1}
    \caption{\label{FigurA}Dette er den første figur.}
  \end{minipage}
  \hfill
  \begin{minipage}{0.3\textwidth}
    {\Huge\bfseries Figur 2}
    \caption{\label{FigurB}Dette er den anden figur.}
  \end{minipage}
  \hfill\
\end{figure}
```

Kommentarer: Med `\begin{figure}` ... `\end{figure}` afgrænses det der skal udgøre den samlede 'float'. Delfigurerne fremstilles i hver sin *minipage* hvor man angiver figurindhold og figurtekst (i eksemplet er der ikke gjort noget for at centrere figurindholdet i forhold til den pågældende minipage). Man kan placere minipage'rne i forhold til hinanden som man ønsker; her bliver de centreret horisontalt ved hjælp af `\hfill`-kommandoerne.

Eksemplet er pænt (måske for pænt) i den forstand at de to delfigurers indhold har samme højde, og deres figurtekst har samme antal linjer, således at minipage'rne får samme højde. Hvis dette ikke havde været tilfældet, skulle man måske angive hvordan minipage'n skulle justeres (se side 28). Hvis figurindholdene ikke er lige høje, kan man sørge for at de bliver det ved at indsætte en strut (se side 93).

Pakker

En *pakke* er i LaTeX-sammenhæng en samling definitioner og kommandoer der supplerer eller modificerer LaTeXs normale opførsel. Eksempelvis er der pakker der sætter LaTeX i stand til at inkludere filer med forskellige slags grafik (f.eks. i eps-format), og pakker der får LaTeX til at benytte andre fonte (f.eks. Adobes PostScript-fonte, eller gamle tyske fonte, eller ekstra matematik-fonte). Pakker benyttes på brugerens foranledning.

Man instruerer LaTeX om at benytte en given pakke ved at skrive en kommando af formen `\usepackage{`*⟨navn⟩*`}` i dokumentets præambel.[2] Eksempel: hvis man ønsker at benytte pakken amsfonts, skal man i præamblen skrive `\usepackage{amsfonts}`.

[2] LaTeX holder selv styr på at den samme pakke ikke bliver indlæst mere end en gang.

Dokumentklasserne fra IMFUFA-LaTeX benytter automatisk et antal pakker der blandt andet sørger for at vælge dansk orddeling.

babel-pakken; orddeling

En del af LaTeXs adfærd skal afhænge af hvilket sprog der skrives på. Det gælder for eksempel benævnelsen af kapitler (skal de hedde Kapitel eller Chapter eller Chapitre eller ...), figurer og tabeller, det gælder den måde datoen skrives på, og det gælder reglerne for orddeling. Og for at gøre det rigtig besværligt, så er det jo ikke altid at hele dokumentet skrives på samme sprog – nogle forfattere ynder at smide om sig med citater på adskillige sprog. Med babel-pakken (Braams, 1999) er der skabt en standard for hvordan dette problem kan løses. Babel-systemet understøtter en lang række især europæiske sprog; af tekniske grunde vil en given LaTeX-installation dog normalt kun understøtte et begrænset antal.

Hvordan gør man?

Det er ganske enkelt at benytte babel, men de nærmere detaljer afhænger af om man anvender en dokumentklasse fra IMFUFA-LaTeX eller fra Standard LaTeX, og af om selve dokumentet skal være ensproget eller flersproget.

Ensprogede dokumenter:

1. Dokumentklasserne fra IMFUFA-LaTeX sørger automatisk for at aktivere babel, og sproget sættes til dansk medmindre man udtrykkeligt beder om noget andet.
2. Hvis man ønsker at udfærdige et engelsk- eller tysksproget dokument i en af IMFUFA-LaTeX-dokumentklasserne (for eksempel fordi de har et pænere layout end Standard LaTeX), så skal man angive `english` eller `german` som en option til dokumentklassen, eksempel:
   `\documentclass[english]{bog}`
3. Hvis man benytter en dokumentklasse fra Standard LaTeX, så skal man i givet fald selv aktivere babel med det ønskede sprog som option, eksempel:
   `\documentclass{book}`
   `\usepackage[danish]{babel}`

Flersprogede dokumenter:

1. Hvis man vil benytte en af de danske dokumentklasser til noget der indeholder både engelsk og dansk tekst, så skal man angive det i dokumentklasseerklæringen, for eksempel sådan:
   `\documentclass[english,danish]{bog}`.
   Derefter kan man skifte mellem sprogene med en kommando af formen `\selectlanguage{`⟨*sprog*⟩`}`.

   Hvis man har en tekst der hovedsagelig er på ét bestemt sprog, for eksempel dansk, kan man med fordel benytte en `otherlanguage*`-konstruktion,

og til korte tekstdele kommandoen `\foreignlanguage`. Der er eksempler på begge dele i Eksempel 4.17.

2. Hvis man benytter en dokumentklasse fra Standard LaTeX, så skal man selv aktivere babel med de ønskede sprog som option, eksempel:
   `\documentclass{book}`
   `\usepackage[english,danish]{babel}`.
   Derefter kan man skifte mellem sprogene på samme måde som ellers.

Det kan tilføjes at der også findes en `otherlanguage`-konstruktion; denne ændrer ikke blot orddelingsreglerne men også de sprogspecifikke tekstkonstanter, f.eks. dem der skriver »Figur« og »Tabel« ved hhv. figurer og tabeller.

Dr Watson forbløffes over Holmes' selektive viden:

My surprise reached a climax, however, when I found incidentally that he was ignorant of the Copernican Theory and of the composition of the Solar System. That any civilized human being in this nineteenth century should not be aware that the earth travelled round the sun appeared to be to me such an extraordinary fact that I could hardly realize it.

»You appear to be astonished,« he said, smiling at my expression of surprise. »Now that I do know it I shall do my best to forget it.«

Derefter forklarer H. hvorfor man ikke skal belaste sin hukommelse med den slags unyttig viden.

A Study In Scarlett, Chapter 2

```
Dr Watson forbløffes over
Holmes' selektive viden:

\begin{otherlanguage*}{english}
My surprise reached a climax,
however, when I found incidentally
that he was ignorant of the
Copernican Theory and of the
composition of the Solar System.
...

\textqm{Now that I do know it
I shall do my best to forget it.}
\end{otherlanguage*}

Derefter forklarer H. hvorfor man
ikke skal belaste sin hukommelse
med den slags unyttig viden.

{\ \hfill \footnotesize
\foreignlanguage{english}{A Study
  In Scarlett, Chapter 2}}
```

EKSEMPEL 4.17

Orddeling

LaTeX har en indbygget orddelingsmekanisme der benytter et sprogspecifikt regelsæt. Desuden er der for hvert sprog en kortere eller længere liste med undtagelser, dvs. ord som orddelingsmekanismen ikke af sig selv kan dele rigtigt (f.eks. et ord som *ukrudtsart*).

Den enkelte bruger kan imidlertid også selv være med til at bestemme hvor ord må deles. Man kan benytte kommandoen `\-` til at markere steder hvor et ord må deles, for eksempel `ukrudts\-art`; bemærk at der ikke sker nogen »indlæring«, men at man skal indsætte `\-` ved hver eneste forekomst af ordet. Hvis der et tale om et ord der optræder mange steder i teksten, kan det derfor betale sig at gøre noget andet, nemlig benytte kommandoen `\hyphenation` til at definere lokale supplerende orddelingsundtagelser. Eksempel:

```
\hyphenation{ukrudts-art ukrudts-ar-ten
  ukrudts-ar-ter ukrudts-ar-terne}
```

(I et flersproget dokument kan man placere en `\selectlanguage`-kommando umiddelbart før `\hyphenation`-kommandoen for at fortælle hvilket sprog orddelingsundtagelserne hører til.)

Undertiden kan man være interesseret i at få at vide hvordan LaTeX i givet fald vil dele et bestemt ord. Til det formål kan man benytte kommandoen `\showhyphens` der *i log-filen* skriver hvordan et eller flere ord kan deles ifølge LaTeX; kommandoen giver ikke noget synligt resultat.

Kapitel 5

Matematik i LaTeX

At skrive tekster der indeholder matematiske argumentationer og udredninger, er ingen let sag. Forfatteren skal først og fremmest gøre sig klart hvad det er der skal siges og formidles, samt sørge for at argumenterne er rigtige og så simple og forståelige som muligt. Dernæst skal det hele omsættes til en læselig skriftlig fremstilling hvor formler og tekst er blandet i det rette forhold og på den rette måde, hvilket blandt andet vil sige at korte formler normalt står inde i teksten, hvorimod længere formler fremhæves.

Formler

For matematiske formler gælder der særlige typografiske regler hvoraf den der først falder i øjnene, er at formler sættes med en særlig kursivskrift: Newtons anden lov opsummeres kort som $F = ma$ (og bestemt ikke som F=ma). I overensstemmelse med den sædvanlige LaTeX-filosofi skal forfatteren ikke bekymre sig om hvad det nu er for en kursivskrift der skal bruges (det er faktisk ikke den samme som tekstkursivskriften), forfatteren skal kun fortælle LaTeX hvor der skiftes fra *tekst-omgivelser* ('text mode') til *matematik-omgivelser* ('math mode') og omvendt.

Formler i teksten

Formler inde i teksten afgrænses af `$ ... $` eller `\( ... \)`:

Newtons anden lov $F = ma$ er et eksempel på en naturlov.	`Newtons anden lov $F=ma$ er et eksempel på en naturlov.`
Newtons anden lov $F = ma$ er et eksempel på en naturlov.	`Newtons anden lov \(F=ma\) er et eksempel på en naturlov.`

EKSEMPEL 5.1

Som man hurtigt vil erfare har tegnet *mellemrum* ingen betydning i formler, det er altså ligegyldigt om man skriver `$F=ma$` eller `$ F = m a $`.

Fremhævede formler

Fremhævede formler ('displayed math') afgrænses af `\[ ... \]`:

EKSEMPEL 5.2

Newtons anden lov

$$F = ma$$

er et eksempel på en naturlov.

```
Newtons anden lov
\[ F=ma \]
er et eksempel på en naturlov.
```

Som det ses, bliver fremhævede formler sat centreret og adskilt fra den almindelige tekst.

Kommandoerne `\displaystyle` og `\textstyle`

Når en formel skal stå i en almindelig tekstlinje, prøver LaTeX temmelig meget på at klemme formlen sammen i højden så den passer til den normale linjeafstand; derfor vil den samme formel få forskelligt udseende afhængigt af om den står som tekstformel eller fremhævet formel. Det illustreres med følgende eksempel (de benyttede matematikkommandoer forklares nærmere i det følgende):

EKSEMPEL 5.3

Summen af en kvotientrække er givet ved $\sum_{k=0}^{n-1} z^k = \frac{1-z^n}{1-z}$ hvilket vises således:

Summen af en kvotientrække er givet ved

$$\sum_{k=0}^{n-1} z^k = \frac{1-z^n}{1-z}$$

hvilket vises således:

```
Summen af en kvotientrække er
givet ved
$ \sum_{k=0}^{n-1}z^{k} =
  \frac{1-z^{n}}{1-z} $
hvilket vises således:

Summen af en kvotientrække er
givet ved
\[ \sum_{k=0}^{n-1}z^{k}=
   \frac{1-z^{n}}{1-z} \]
hvilket vises således:
```

Med kommandoerne `\displaystyle` og `\textstyle` kan man tvinge LaTeX til at benytte de regler der gælder for henholdsvis fremhævede formler og tekstformler, også selv om det egentlig ikke hører til i sammenhængen:

EKSEMPEL 5.4

Summen af en kvotientrække er givet ved $\displaystyle\sum_{k=0}^{n-1} z^k = \frac{1-z^n}{1-z}$ hvilket vises således:

Summen af en kvotientrække er givet ved

$$\textstyle\sum_{k=0}^{n-1} z^k = \frac{1-z^n}{1-z}$$

hvilket vises således:

```
Summen af en kvotientrække er
givet ved
$\displaystyle
  \sum_{k=0}^{n-1}z^{k} =
  \frac{1-z^{n}}{1-z} $
hvilket vises således:

Summen af en kvotientrække er
givet ved
\[\textstyle
   \sum_{k=0}^{n-1}z^{k}=
   \frac{1-z^{n}}{1-z} \]
hvilket vises således:
```

De grundlæggende ideer

Det er et grundlæggende princip i LaTeX at al input skal foregå ved hjælp af de bogstaver, cifre og tegn der nu engang er på almindelige computer-tastaturer, og heraf følger at beskrivelsen af matematiske formler skal ske ved hjælp af kommandoer. De fleste af LaTeXs matematikkommandoer har »bogstav-navne«, der er valgt som (en forkortelse af) symbolets (engelske) navn, eksempelvis er kommandoen for et sumtegn ($\sum$) `\sum` og kommandoen for uendelig-symbolet (∞) `\infty`; visse ting angives dog med særlige tegn. Vi præsenterer nu de vigtigste matematikkommandoer og -kommandotyper.

Fod- og toptegn

Fodtegn ('subscript') og toptegn ('superscript') markeres med _ (understreg) og ^ (cirkumfleks); det der skal stå som fodtegn eller toptegn, omgives af { }:

$z^2 \quad e^{i(s+t)}$	`z^{2}  e^{i(s+t)}`
$n_k \quad x_{n_k}$	`n_{k}  x_{n_{k}}`

EKSEMPEL 5.5

Det er ikke syntaktisk forkert at udelade { }, men se hvad der sker:

$e^i t \quad e^{it}$	`e^it  e^{it}`

EKSEMPEL 5.6

Sum, produkt, integral

Sumtegn, *produkttegn* og *integraltegn* fremstilles med kommandoerne `\sum`, `\prod` og `\int`; eventuelle grænser angives som fod- og toptegn:

$\sum_{n=0}^{\infty} z^n = \frac{1}{1-z}$	`\sum_{n=0}^{\infty}z^{n} = \frac{1}{1-z}`
$\prod_{k=1}^{n} k = n!$	`\prod_{k=1}^{n} = n!`
$\int_0^{+\infty} e^{-t} dt = 1$	`\int_{0}^{+\infty} e^{-t} dt =1`

EKSEMPEL 5.7

Som man kan se i Eksempel 5.3, ændrer sumtegnet størrelse afhængigt af om det står i en fremhævet formel eller i en tekstformel. Det samme gælder for produkttegn og integraltegn samt visse andre tegn, se side 57.

Mellemrum

I matematiksammenhæng skal en bogstavsekvens som for eksempel *sut* sædvanligvis ikke opfattes som ordet bestående af bogstaverne s, u og t, men derimod som produktet af de tre størrelser s, u og t. Da man således ikke opererer med

ord, men med symboler bestående af enkeltbogstaver, er tegnet *mellemrum* blevet tildelt en anden betydning i matematiksammenhæng end i tekstsammenhæng:

EKSEMPEL 5.8

Når man skriver formler, bliver mellemrum ikke til synlige mellemrum: $sut = sut$.	`Når man skriver formler, bliver mellemrum ikke til synlige mellemrum: $sut = s u t$.`

Undertiden kan man godt ønske sig andre afstande mellem enkeltdelene i en formel end dem som LaTeX laver af sig selv, det kan fremme læseligheden. Således burde der i integralet i Eksempel 5.7 være større afstand mellem e^{-t} og dt end der faktisk er.

Der findes forskellige kommandoer der kan indsætte mellemrum i formler. Her er en oversigt over disse kommandoer og deres virkning; mellemrummets størrelse er den vandrette afstand mellem de to sorte rektangler:

kommando	eksempel	kommentar
		intet mellemrum
`\!`		negativt mellemrum
`\,`		lille mellemrum
`\:`		middel mellemrum
`\␣`		et ord-mellemrum
`\enspace`		en halv geviert
`\quad`		en geviert
`\qquad`		to gevierter

(En *geviert* er lig bredden af et M i den aktuelle skrift.)

Med undtagelse af `\!` og `\:` kan disse kommandoer også anvendes i tekstsammenhæng.

EKSEMPEL 5.9

Lysets hastighed i vacuum er 299 792 458 m/s.	`Lysets hastighed i vacuum er 299\,792\,458 m/s.`
Lysets hastighed i vacuum er 299 792 458 m/s.	`Lysets hastighed i vacuum er $299\,792\,458$ m/s.`

Almindelige regneoperationer

Almindelige regnestykker skrives stort set som man umiddelbart skulle tro. Som multiplikationstegn bruges alt efter omstændighederne eller sædvane eller smag og behag enten ingenting eller en centreret prik eller tegnet $\times$:

EKSEMPEL 5.10

$ab = a \cdot b$	`$ab=a \cdot b$`
$3 \times 5 = 15$	`$3 \times 5 = 15$`
$(a+b)/c = a/c + b/c$	`$(a+b)/c=a/c + b/c$`

EKSEMPEL 5.15

$\frac{1}{\sqrt{x(1-x)}}$	`$\frac{1}{\sqrt{x(1-x)}}$`
$\frac{1}{\:\sqrt{x(1-x)}\:}$	`$\frac{1}{\:\sqrt{x(1-x)}\:}$`
$\frac{1}{\ \sqrt{x(1-x)}\ }$	`$\frac{1}{\ \sqrt{x(1-x)}\ }$`

Navngivne funktioner og operatorer

En del matematiske funktioner og operatorer har standardbetegnelser som er forkortelser af deres navne, f.eks. sin, log, lim sup. Disse må *under ingen omstændigheder* skrives med kursiv, og det sker heller ikke hvis man bærer sig rigtigt ad, dvs. hvis man skriver betegnelserne som LaTeX-kommandoer, f.eks.

EKSEMPEL 5.16

$\exp(ix) = \cos x + i \sin x$	`\exp(ix)=\cos x + i \sin x`

Visse af symbolerne har den egenskab, at et eventuelt fodtegn står *under* symbolet i fremhævede formler:

EKSEMPEL 5.17

$\lim_{n\rightarrow\infty} x_{n}$	`\lim_{n\rightarrow\infty} x_{n}`

På side 57 er en liste over samtlige navngivne funktioner.

Differentiation

EKSEMPEL 5.18

$f'' = (f')'$	`$f''=(f')'$`
$a = \dot v = \ddot x$	`$a = \dot v = \ddot x$`
$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2}$	`$\frac{\partial f}{\partial t} = \frac{\partial^2 f}{\partial x^2}$`
∇f	`$\nabla f$`

Relationer

LaTeX har en lang række relationssymboler. Her er nogle der vedrører tal:

EKSEMPEL 5.19

$x = x$, $x^2 \geq 0$, $-x^2 \leq 0$, $5 > 3$, $3 < 5$, $3 \neq 5$.	`$x=x$, $x^2 \geq 0$, $-x^2 \leq 0$, $5 > 3$, $3 < 5$, $3 \neq 5$.`

Her er nogle der vedrører mængder:

$x \in A$, $A \ni x$,	`$x \in A$, $A \ni x$,`
$A \subset B$, $B \supset A$,	`$A \subset B$, $B \supset A$,`
$A \subseteq B$, $B \supseteq A$.	`$A \subseteq B$, $B \supseteq A$.`

EKSEMPEL 5.20

Man kan negere relationerne med kommandoen `\not`:

$5 \not< 3$, $8 \not\geq 9$,	`$5 \not< 3$, $8 \not\geq 9$,`
$x \notin A$, $B \not\subset A$.	`$x \notin A$, $B \not\subset A$.`

EKSEMPEL 5.21

Hvis man vil være bagvendt, kan man bruge IMFUFA-LaTeX-kommandoen `\ton`:

$5 \ntoin 3$, $8 \ntoeq 9$,	`$5 \ntoin 3$, $8 \ntoeq 9$,`
$x \ntoin A$, $B \ntosubset A$.	`$x \ntoin A$, $B \ntosubset A$.`

EKSEMPEL 5.22

I oversigterne side 57ff findes talrige andre relationssymboler.

Græske bogstaver

Kommandoerne for græske bogstaver til matematikbrug er nemme, de er ganske enkelt bogstavets navn. Man skal dog være opmærksom på at navnene skal staves på engelsk, således at kommandoen for α er `\alpha`, ikke `\alfa`. På side 63 er der en liste over de græske bogstaver og deres kommandoer.

Tre prikker

Ved opremsninger hvor man erstatter mellemliggende elementer med prikker, bruges som bekendt *tre* prikker, hverken mere eller mindre. De frembringes med kommandoen `\ldots` (der også kan anvendes i tekstsammenhæng):

$1+2+\ldots+n = n(n+1)/2$.	`$1+2+\ldots+n = n(n+1)/2$.`
Lad $z_1, z_2, \ldots, z_n$ være komplekse tal.	`Lad $z_1,z_2,\ldots,z_n$ være komplekse tal.`

EKSEMPEL 5.23

Der findes fire forskellige udgaver af »tre prikker«:

$\ldots$ er prikker på linjen,	`$\ldots$ er prikker på linjen,`
$\cdots$ er centrerede prikker,	`$\cdots$ er centrerede prikker,`
$\vdots$ er vertikale prikker,	`$\vdots$ er vertikale prikker,`
$\ddots$ er diagonale prikker.	`$\ddots$ er diagonale prikker.`

EKSEMPEL 5.24

De tre af dem benyttes i Eksempel 5.43 på side 50.

Parenteser og deslige

Matematikere og matematik-brugere betjener sig af et større udvalg af parenteser og parentesagtige symboler:

EKSEMPEL 5.25

(x) er x med parenteser om	`$(x)$ er $x$ med parenteser om`
$[x]$ er den hele del af x	`$[x]$ er den hele del af $x$`
$\{x\}$ er mængden bestående af x	`$\{x\}$ er mængden bestående af $x$`
$\lvert x\rvert$ er den numeriske værdi af x	`$\lvert x\rvert$ er den numeriske værdi af $x$`
$\lVert x\rVert$ er normen af x	`$\lVert x\rVert$ er normen af $x$`
$\langle x\rangle$ er middelværdien af x – hvis man er fysiker	`$\langle x \rangle$ er middelværdien af $x$ -- hvis man er fysiker.`

Alle disse parenteser samt visse andre symboler, se side 58, kan bringes til at skifte størrelse; derved kan man forøge såvel formlernes æstetiske kvaliteter som deres læselighed. Her er for eksempel to forskellige måder at skrive det samme matematiske udtryk på:

$$(\int_0^1 t(1-t)\,dt)^2 \quad = \quad \left(\int_0^1 t(1-t)\,dt\right)^2 .$$

Udgaven til højre er klart at foretrække; den fremstilles således:

EKSEMPEL 5.26

$$\left(\int_0^1 t(1-t)\,dt\right)^2$$

```
\[ \left( \int_{0}^{1} t(1-t)\,dt
   \right)^{2} \]
```

De forstørrede parenteser er blevet til ved at man i stedet for (og) har skrevet `\left(` og `\right)`. Man kan have flere `\left` `\right` konstruktioner inden i hinanden:

EKSEMPEL 5.27

$$\left(\sum_{i=1}^{n-1}\left|\frac{z_i}{z_{i+1}}\right|\right)^2$$

```
\[ \left( \sum_{i=1}^{n-1}
     \left| \frac{z_{i}}{z_{i+1}}
     \right|
   \right)^{2} \]
```

LaTeX er ligeglad med om parenteserne matcher i matematisk forstand, så man kan sagtens skrive f.eks.

EKSEMPEL 5.28

$$\left\}\int_0^1 t(1-t)\,dt\right|^2$$

```
\[ \left\} \int_{0}^{1} t(1-t)\,dt
   \right|^{2} \]
```

Derimod *skal* parenteserne matche i LaTeX-forstand, dvs. hver `\left` skal have en tilsvarende `\right`. For at opnå dette kan man undertiden have stor glæde af en *usynlig* parentes der skrives som `\left.` eller `\right.`, alt efter behov:

EKSEMPEL 5.29

$$\left.\frac{\partial x}{\partial t}\right|_{t=0}$$

```
\[ \left.
     \frac{\partial x}{\partial t}
   \right|_{t=0} \]
```

Se også Eksempel 5.42, 5.43 og 5.44 med hhv. matricer og en Tuborg-funktion; i Eksempel 9.9 på side 93 vises hvordan man kan fremtvinge bestemte størrelser parenteser.

Pile

LaTeX har et stort udvalg af pile. Her er nogle få eksempler:

Funktionen $x \mapsto x^2$ er kontinuert.	`Funktionen $x \mapsto x^2$ er kontinuert.`
$\ln x \to 0 \iff x \to 1$.	`$\ln x \rightarrow 0 \iff x \rightarrow 1$.`

EKSEMPEL 5.30

IMFUFA-LaTeX har en særlig kommando `\limarrow` der kan bruges til grænseovergange:

$\ln x \xrightarrow[x\to 1]{} 0$	`$\ln x\limarrow{x\rightarrow 1} 0$`

EKSEMPEL 5.31

Der er en oversigt over pile på side 62.

Accenter

Der er særlige kommandoer til accenter i matematiksammenhæng (se også Eksempel 5.33):

$\acute{x}$ $\grave{x}$ $\hat{x}$ $\tilde{x}$ $\bar{x}$	`$\acute{x} \grave{x} \hat{x} \tilde{x} \bar{x}`
$\check{x}$ $\breve{x}$ $\mathring{x}$ $\dot{x}$ $\ddot{x}$ $\vec{x}$	`\check{x} \breve{x} \mathring{x} \dot{x} \ddot{x} \vec{x}$`

EKSEMPEL 5.32

Ting over og under formler

Man kan sætte lange hatte, tilder, streger og pile over formler:

$\widehat{x+y}$ $\widetilde{x+y}$ $\overline{x+y}$	`$\widehat{x+y} \widetilde{x+y} \overline{x+y}`
$\underline{x+y}$ $\overrightarrow{x+y}$ $\overleftarrow{x+y}$	`\underline{x+y} \overrightarrow{x+y} \overleftarrow{x+y}$`

EKSEMPEL 5.33

Man kan også have glæde af kommandoerne `\underbrace` og `\overbrace`:

$1,2,\underbrace{3,3,\ldots,3}_{n},4$	`$1,2, \underbrace{3,3,\ldots,3}_{n},4$`
$9,8,\overbrace{7,7,\ldots,7}^{k},6$	`$9,8, \overbrace{7,7,\ldots,7}^{k},6$`

EKSEMPEL 5.34

Symboler for talmængderne

IMFUFA-LaTeX indeholder kommandoerne `\N`, `\Z`, `\Q`, `\R` og `\C` der fremstiller symboler for de almindelige talmængder[1]:

EKSEMPEL 5.35

De naturlige tal $\mathbf{N}$	`De naturlige tal $\N$`
De hele tal $\mathbf{Z}$	`De hele tal $\Z$`
De rationale tal $\mathbf{Q}$	`De rationale tal $\Q$`
De reelle tal $\mathbf{R}$	`De reelle tal $\R$`
De komplekse tal $\mathbf{C}$	`De komplekse tal $\C$`
$(x,y) \in \mathbf{R}^2$	`$(x,y)\in\R^{2}$`

Det er god LaTeX-filosofi at opretholde en fast sammenhæng mellem LaTeX-symbolerne og deres *matematiske* indhold, og så justere symbolernes *typografiske* indhold efter ønske. Det betyder at hvis man af en eller anden grund hellere vil skrive de reelle tal som $\mathbb{R}$ (selv om dette tegn er grimt og ikke passer ind i den typografiske sammenhæng), så skal man bibeholde kommandoen `\R` for de reelle tal, men omdefinere dens typografiske indhold, se punkt 2 på side 85.

Diverse kommandoer i IMFUFA-LaTeX

IMFUFA-LaTeX definerer en kommando `\binom` der laver binomialkoefficienter,

EKSEMPEL 5.36

En binomialkoefficient $\binom{n}{k}$.	`En binomialkoefficient $\binom{n}{k}$.`
$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$	`\[ \binom{n}{k} = \frac{n!}{k!\,(n-k)!} \]`

og en kommando `\ntup` der laver n-tupler:

EKSEMPEL 5.37

Lad $X_1, X_2, \ldots, X_k$ være uafhængige stokastiske variable.	`Lad $\ntup{X}{k}$ være uafhængige stokastiske variable.`

Desuden er der sandsynlighedsregnings-kommandoerne `\Pr`, `\Exp` og `\Var`:

EKSEMPEL 5.38

$\mathrm{P}(A \cap B) = \mathrm{P}(A) \cdot \mathrm{P}(B)$ hvis A og B er uafhængige.	`$\Pr(A\cap B)=\Pr(A)\cdot\Pr(B)$ hvis $A$ og $B$ er uafhængige.`
$\mathrm{Var}X = \mathrm{E}(X - \mathrm{E}X)^2$	`$\Var X=\Exp(X-\Exp X)^{2}$`

Diverse tegn

Der findes en lang række tegn, f.eks. $\forall$ (`\forall`), $\exists$ (`\exists`), $\neg$ (`\neg`), $\emptyset$ (`\emptyset`), $\hbar$ (`\hbar`). Se i øvrigt oversigterne side 57ff.

[1] Kommandoerne er defineret ved hjælp af `\mathbf`-kommandoen, se Eksempel 5.59.

Tekst i formler

Undertiden har man brug for at indsætte tekst i formler. Det kan gøres ved hjælp af kommandoerne til tekst-skriftvalg (Tabel 3.1 side 23). Her vises det med `\textrm`; husk at betydningen af tegnet mellemrum er forskelligt i matematik- og i tekstsammenhæng.

$x^2 \geq 0$ for alle x	`$x^2 \geq 0 \textrm{ for alle }x$`
Gnidningskraften $F_{\textrm{gnidn}}$ er 0	`Gnidningskraften` `  $F_{\textrm{gnidn}}$ er $0$`
1 Å $= 10^{-10}$ m	`$1\textrm{ Å}=10^{-10}\textrm{ m}$`

EKSEMPEL 5.39

Som det ses, skifter teksten ikke størrelse når den står som fodtegn, og det er jo ikke pænt. Tingene ændrer sig hvis man benytter pakken amstext (eller amsmath); så bliver resultatet sådan her:

$x^2 \geq 0$ for alle x	`$x^2 \geq 0 \textrm{ for alle }x$`
Gnidningskraften $F_{\textrm{gnidn}}$ er 0	`Gnidningskraften` `  $F_{\textrm{gnidn}}$ er $0$`
1 Å $= 10^{-10}$ m	`$1\textrm{ Å}=10^{-10}\textrm{ m}$`

EKSEMPEL 5.40

Symbolerne m og Å i ovenstående formler er sådan set ikke tekst, men netop symboler. Derfor vil det være mere rigtigt at skrive dem som `\mathrm{m}` og `\mathrm{Å}` (kommandoen `\mathrm` omtales side 54). Der er ingen problemer med `\mathrm{m}`, men `\mathrm{Å}` går galt fordi LaTeXs matematikskrifter desværre stadig (foråret 1999) ikke indeholder det eksotiske bogstav å.

array-konstruktioner

Man kan benytte `array`-konstruktioner for at fremstille skematiske opstillinger, for eksempel matricer. Der er mange ligheder mellem `array`- og `tabular`-konstruktioner (se side 25): Efter `\begin{array}` skriver man en liste af bogstaver (som kan være `l`, `r` eller `c`) svarende til det ønskede antal søjler; de enkelte felter i hver linje adskilles af `&` og de enkelte linjer adskilles af `\\`. – Et `array` kan kun optræde i matematiksammenhæng.

Et array med tre søjler og to rækker:	`Et array med tre søjler` `og to rækker:`

$$\begin{array}{ccl} x+y & z & a+\frac{1}{3} \\ i & K^2 & A\cup B\cup C \end{array}$$	`\[`
	`  \begin{array}{ccl}`
	`  x+y & z  & a+\frac{1}{3} \\`
	`  i  & K^2& A\cup B\cup C`
	`  \end{array}`
	`\]`

EKSEMPEL 5.41

Som eksemplet antyder bliver de enkelte felter sat efter 'textstyle'-reglerne (jf. side 40); det kan man lave om på ved at indsætte en `\displaystyle` i de felter hvor det skal ændres.

En typisk anvendelse af `array` er til matricer (her får vi brug for `\left` `\right`-konstruktionen, se side 45):

EKSEMPEL 5.42

En 3 × 3-matrix:

$$A=\left(\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{array}\right)$$

```
En $3\times 3$-matrix:
\[ A=\left(\begin{array}{ccc}
    a_{11} & a_{12} & a_{13}\\
    a_{21} & a_{22} & a_{23}\\
    a_{31} & a_{32} & a_{33}
  \end{array}\right)\]
```

Til en $n \times k$-matrix får vi brug for forskellige typer prikker (Eksempel 5.24):

EKSEMPEL 5.43

En $n \times k$-matrix:

$$A=\left(\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nk} \end{array}\right)$$

```
En $n\times k$-matrix:
\[ A=\left(\begin{array}{cccc}
 a_{11} & a_{12} &\ldots & a_{1k}\\
 a_{21} & a_{22} &\ldots & a_{2k}\\
 \vdots & \vdots &\ddots & \vdots\\
 a_{n1} & a_{n2} &\ldots & a_{nk}
 \end{array}\right)\]
```

Et `array` kan stå alle steder hvor der kan stå et »almindeligt« matematiksymbol, her benyttes det for eksempel til at konstruere en Tuborgfunktion:

EKSEMPEL 5.44

Funktionen sgn er defineret ved

$$\mathrm{sgn}(x)=\left\{\begin{array}{rl} 1 & \text{hvis } x>0 \\ 0 & \text{hvis } x=0 \\ -1 & \text{hvis } x<0 \end{array}\right.$$

```
Funktionen $\mathrm{sgn}$
er defineret ved
\[ \mathrm{sgn}(x) =
    \left\{\begin{array}{rl}
      1 &\textrm{ hvis }x>0\\
      0 &\textrm{ hvis }x=0\\
     -1 &\textrm{ hvis }x<0
    \end{array}\right. \]
```

Opstillinger og nummererede formler

Hidtil har vi omtalt formler der står i den almindelige tekst, samt fremhævede formler der fylder én linje. Imidlertid har man ofte brug for at skulle præsentere længere matematiske omskrivninger der ønskes fordelt over flere linjer sådan at lighedstegnene (eller ulighedstegnene eller hvad det nu er) står pænt under hverandre; hertil bruges `eqnarray*`- eller `eqnarray`-konstruktionen.

Opstillinger med `eqnarray*`

Et `eqnarray*` er en formelopstilling der strækker sig over flere linjer, og hvor hver linje består af *tre felter*: venstrefeltet, midterfeltet og højrefeltet. Man

bruger tegnet & til at adskille felterne og tegnet \\ til at adskille de enkelte *formellinjer* (en formellinje er en linje i den resulterende formel; den LaTeX-kode der frembringer en formellinje, kan sagtens strække sig over adskillige linjer i input-filen).

Der gælder at

$$\begin{aligned}(a+b)^2 &= a^2+2ab+b^2\\ &\leq a^2+2|ab|+b^2\\ &= (|a|+|b|)^2\end{aligned}$$

for alle reelle tal a og b.

```
Der gælder at
\begin{eqnarray*}
  (a+b)^2 & = & a^2 + 2ab + b^2 \\
      &\leq & a^2 + 2|ab| + b^2 \\
      & = & (|a|+|b|)^2
\end{eqnarray*}
for alle reelle tal $a$ og $b$.
```

EKSEMPEL 5.45

Bemærk venligst at der skal være præcis to &'er i hver linje, og at der *ikke* skal \\ efter den sidste linje.

Som det ses kan et felt godt være tomt.

Opstillingen indrettes på den måde at midterfelterne står under hverandre centreret, venstrefelterne står under hverandre højrejusteret, og højrefelterne står under hverandre venstrejusteret (dvs. yderfelterne rykkes ind mod midterfeltet). I øvrigt sættes midterfeltet efter 'textstyle'-reglerne, medens yderfelterne sættes som 'displaystyle' (jf. side 40).

Et `eqnarray*` kan udmærket strække sig over flere sider. Derimod opstår der problemer hvis nogle af linjerne er meget lange. Hvis den første venstreside i et `eqnarray*` er meget lang, kan man benytte `\lefteqn` for at få en »fritstående« venstreside:

$$\begin{aligned}&a+b+c+d+e+f\\ &= g+h\\ &= x^2\end{aligned}$$

```
\begin{eqnarray*}
  \lefteqn{a+b+c+d+e+f} \\
    & = & g+h \\
    & = & x^2
\end{eqnarray*}
```

EKSEMPEL 5.46

Nummererede formler

Man kan få LaTeX til at nummerere fremhævede formler, både enlinjesformler og linjerne i større opstillinger.

Man får nummereret en enlinjesformel ved at udskifte kommandoerne `\[` `\]` med `\begin{equation}` `\end{equation}`; formelnumre skrives traditionelt altid til højre for formlen (hvilket ikke ser pænt ud i eksempelkasserne):

Den berømte formel

$$E = mc^2 \tag{5.1}$$

```
Den berømte formel
\begin{equation}
  E=mc^2
\end{equation}
```

EKSEMPEL 5.47

Man får nummereret formlerne i et `eqnarray*` ved at skrive `eqnarray` i stedet for `eqnarray*`; her viser vi det med Eksempel 5.45:

EKSEMPEL 5.48

Der gælder at

$$\begin{aligned}(a+b)^2 &= a^2+2ab+b^2 && (5.2)\\ &\leq a^2+2|ab|+b^2 && (5.3)\\ &= (|a|+|b|)^2 && (5.4)\end{aligned}$$

for alle reelle tal a og b.

```
Der gælder at
\begin{eqnarray}
  (a+b)^2 & = & a^2 + 2ab + b^2 \\
      &\leq & a^2 + 2|ab| + b^2 \\
      & = & (|a|+|b|)^2
\end{eqnarray}
for alle reelle tal $a$ og $b$.
```

I eksemplet er det klart urimeligt at nummerere *alle* linjerne; man kan forhindre nummereringen af en linje ved at anbringe kommandoen `\nonumber` et eller andet sted på linjen (husk at de enkelte linjer adskilles af `\\`). Hvis vi i Eksempel 5.48 kun vil have den sidste linje nummereret, skal der placeres en `\nonumber`-kommando på de to andre:

EKSEMPEL 5.49

Der gælder at

$$\begin{aligned}(a+b)^2 &= a^2+2ab+b^2\\ &\leq a^2+2|ab|+b^2\\ &= (|a|+|b|)^2 && (5.5)\end{aligned}$$

for alle reelle tal a og b.

```
Der gælder at
\begin{eqnarray}
  \nonumber
  (a+b)^2 & = & a^2 + 2ab + b^2 \\
  \nonumber
      &\leq & a^2 + 2|ab| + b^2 \\
      & = & (|a|+|b|)^2
\end{eqnarray}
for alle reelle tal $a$ og $b$.
```

Man henviser til de nummererede formler med `\ref` og `\label` sådan som det er beskrevet på side 14ff. En label skal stå på den formellinje den hører til (husk stadig at de enkelte formellinjer adskilles af `\\`).

Der er tradition for at formelnumre står i runde parenteser, ikke bare i formlerne men også når man henviser til dem. Imidlertid skriver LaTeX ikke parenteser om de numre der frembringes af `\ref`; man må så selv sætte parenteserne hver gang, eller man kan definere sig en kommando `\refeqn` der gør det (se punkt 6 på side 86).

EKSEMPEL 5.50

Dette er formel (5.6):

$$E = mc^2 \qquad (5.6)$$

Ovenstående er formel (5.6).

```
Dette er formel (\ref{formel1}):
\begin{eqnarray}
  E&=&mc^2 \label{formel1}
\end{eqnarray}
Ovenstående er formel
\refeqn{formel1}.
```

Sætning og bevis

Matematiske sætninger indeholder ofte sætninger og beviser. Disse får hensigtsmæssige typografiske rammer med konstruktionerne `theorem` og `proof`:

En vigtig sætning:

Sætning 5.1
Et polynomium af grad n har n komplekse rødder.

```
En vigtig sætning:
\begin{theorem}
  Et polynomium af grad $n$
  har $n$ komplekse rødder.
\end{theorem}
```

EKSEMPEL 5.51

og her er beviset:

Bevis
Beviset forbigås. □

```
og her er beviset:
\begin{proof}
  Beviset forbigås.
\end{proof}
```

EKSEMPEL 5.52

Undertiden kan det være hensigtsmæssigt at præcisere hvad det er for en sætning der bevises; det kan man skrive i `[ ]` efter `\begin{proof}`:

og her er beviset:

Bevis for Sætning 5.1:
Beviset forbigås. □

```
og her er beviset:
\begin{proof}[for
    Sætning \ref{algft}:]
  Beviset forbigås.
\end{proof}
```

EKSEMPEL 5.53

Man kan navngive en sætning ved at angive et navn i `[ ]`; som det ses skal navnet ikke være for langt:

En vigtig sætning:

Sætning 5.2 (Algebraens Fundamentalsætning)
Et polynomium af grad n har n komplekse rødder.

```
En vigtig sætning:
\begin{theorem}[Algebraens
Fundamentalsætning]
  Et polynomium af grad $n$
  har $n$ komplekse rødder.
\end{theorem}
```

EKSEMPEL 5.54

Ofte opererer man ikke blot med sætninger, men også med f.eks. lemmaer, hovedsætninger og korollarer. Så kan man (i præamblen) definere nye såkaldt »sætningslignende« konstruktioner ved hjælp af kommandoen `\newtheorem`. Vi kan eksempelvis definere en `maintheorem`-konstruktion ved at skrive

```
\newtheorem{maintheorem}{Hovedsætning}
```

Den benyttes således:

Der gælder

Hovedsætning 1
Dette er en hovedsætning.

```
Der gælder
\begin{maintheorem}
  Dette er en hovedsætning.
\end{maintheorem}
```

EKSEMPEL 5.55

Disse hovedsætninger bliver nummereret fortløbende; hvis man hellere vil have dem nummereret inden for kapitler, skal man definere dem som

```
\newtheorem{maintheorem}{Hovedsætning}[chapter]
```

Vi kan også definere eksempelvis lemmaer der nummereres med den samme tæller som sætningerne:

```
\newtheorem{lemma}[theorem]{Lemma}
```

Det virker sådan:

EKSEMPEL 5.56

Først vises
 	
Lemma 5.3
Dette er et lemma.
	`Først vises`
	`\begin{lemma}`
	`  Dette er et lemma.`
	`\end{lemma}`

Det er endvidere muligt at definere nye »bevis-lignende« strukturer, f.eks.

```
\newproof{remark}{Bemærkning}
```

Det viser vi ingen eksempler på, men slutter med et par bemærkninger:
1. babel-systemet sørger for at de prædefinerede `theorem` og `proof` passer til det valgte sprog (jf. side 36).
2. IMFUFA-LaTeX benytter ikke Standard LaTeX-udgaven af `theorem`, men derimod den udgave der findes i theorem-pakken (Mittelbach, 1989-95).
3. `proof`-konstruktionen og kommandoen `\newproof` findes kun i IMFUFA-LaTeX.

Skriftvalg i matematiksammenhæng

I tekstsammenhæng, herunder også når man vil skrive tekst i formler, vælger man skrift med en kommando af formen `\textxx` (jf. Tabel 3.1 på side 23 samt Eksempel 5.40), i matematiksammenhæng foregår det med en kommando af formen `\mathxx`. – Bemærk at ingen af matematikskrifterne har bogstaverne æ, ø og å.

- `\mathit`

  I matematik er der den konvention at når man skriver bogstaver ved siden af hinanden, så er der normalt et underforstået multiplikationstegn imellem dem, for eksempel betyder $fisk$ produktet af de fire størrelser f, i, s og k. (Konventionerne er mere indviklede end som så; hvis man skriver $\sin x$, så udgør de tre første bogstaver ét symbol, hvilket ses af at de ikke er sat med kursiv.)

  Den normale matematikskrift ('math italic') er indrettet efter denne konvention; det kan blandt andet ses på størrelsen af mellemrummene mellem bogstaverne og på de manglende ligaturer. Dette er tekstkursiv: *fisk* og dette er matematikskrift: $fisk$.

  Hvis man ønsker at operere med en matematikvariabel der betegnes med et symbol bestående af flere bogstaver, kan man benytte kommandoen `\mathit`:

Tekstkursiv: *fisk*.
Fire faktorer: $fisk = fiks$.
En variabel: *fisk*.
Endnu en variabel: *fedefisk*.
$v = \ln fisk$.
	`Tekstkursiv: \textit{fisk}.`
	`Fire faktorer: $fisk=fiks$.`
	`En variabel: $\mathit{fisk}$.`
	`Endnu en variabel:`
	`  $\mathit{fede fisk}$.`
	`$v=\ln \mathit{fisk}$.`

EKSEMPEL 5.57

Eksemplet illustrerer også at tegnet mellemrum tolkes på en anden måde i matematiksammenhæng.

- `\mathrm`

I matematik har man den konvention at flerbogstavsnavne på funktioner skrives med antikva. LaTeX har standardkommandoer for en lang række af disse funktioner (se side 57); hvis man har brug for flere, kan man benytte kommandoen `\mathrm`:

Hvis $\mathrm{corr}(X,Y) = 0$, er X og Y ukorrelerede.	`Hvis $\mathrm{corr}(X,Y)=0$,` `er $X$ og $Y$ ukorrelerede.`

EKSEMPEL 5.58

- `\mathbf`

Hvis man vil have fede symboler i formlerne, skal det som hovedregel være en fed udgave af den normale matematikskrift, se hvordan i Eksempel 5.66.

Hvis man har brug for et fedt bogstav i antikvaskriften, kan man bruge kommandoen `\mathbf`:

For $x \in \mathbf{H}$ gælder	`For $x\in\mathbf{H}$ gælder`

EKSEMPEL 5.59

Symbolerne $\mathbf{N}$, $\mathbf{Z}$, $\mathbf{Q}$, $\mathbf{R}$ og $\mathbf{C}$ (Eksempel 5.35) er defineret ved hjælp af `\mathbf`.

- `\mathsf`

En blokskrift til matematik:

For $x \in \mathsf{A}$ gælder	`For $x\in\mathsf{A}$ gælder`

EKSEMPEL 5.60

- `\mathtt`

En skrivemaskineskrift til matematik:

For $x \in \mathtt{A}$ gælder	`For $x\in\mathtt{A}$ gælder`

EKSEMPEL 5.61

- `\mathcal`

  En »skriveskrift« til matematik; den har kun store bogstaver.

EKSEMPEL 5.62

For $x \in \mathcal{A}$ gælder	`For $x\in\mathcal{A}$ gælder`

- `\mathbb`

  De såkaldte 'blackboard bold' er defineret i pakken amsfonts. Der er kun store bogstaver.

EKSEMPEL 5.63

$\mathbb{ABC}$	`$\mathbb{ABC}$`

- `\mathfrak`

  En frakturskrift (»krøllede bogstaver«) til matematikbrug. Den er defineret i pakken amsfonts.

EKSEMPEL 5.64

$\mathfrak{xyz\ XYZ}$	`$\mathfrak{xyz\ XYZ}$`

Bemærk at `\mathfrak` kun er til matematikbrug. Hvis man vil skrive tekst i frakturskrift, så har LaTeX bedre tilbud.

Fede formler og fede enkeltsymboler

Kommandoen `\mathversion{bold}` bevirker at alt efterfølgende matematik bliver sat med en fed skrift; man skifter tilbage med `\mathversion{normal}`. Den fede skrift kan man have brug for i overskrifter eller til transparenter.

EKSEMPEL 5.65

$\boldsymbol{y_\nu = X\beta \in \mathcal{Y}}$	`\mathversion{bold}` `$y_{\nu}=X\beta \in\mathcal{Y}$`
$y_\nu = X\beta \in \mathcal{Y}$	`\mathversion{normal}` `$y_{\nu}=X\beta \in\mathcal{Y}$`

Bemærk at `\mathversion{...}` skal stå i *tekst*sammenhæng (og altså ikke inde i en formel).

Hvis det kun er enkelte symbol der skal skrives med fed skrift, kan kommandoen `\bm` benyttes. Kommandoen `\bm` er defineret i pakken bm; det anbefales at man indlæser denne pakke *efter* eventuelle andre pakker der laver om på skriftvalget (Carlisle, 1999).

EKSEMPEL 5.66

$\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$	`$\bm{x}` `=(x_1,x_2,\ldots,x_n)$`
$2t\boldsymbol{\alpha}$	`$2t\bm{\alpha}$`

Hvis man skal bruge det samme fede symbol mange gange, kan det betale sig at definere det som en særlig kommando. Det gør man med `\bmdefine` (normalt i præamblen):

Nu kan vi skrive $\Phi + \boldsymbol{\Phi}$.	`\bmdefine{\FFi}{\bm{\Phi}}` `Nu kan vi skrive $\Phi+\FFi$.`

EKSEMPEL 5.67

Skematiske oversigter

I dette afsnit gives oversigter over de forskellige især matematiske symboler der er til rådighed i LaTeX. Symbolerne er søgt grupperet efter betydning eller i visse tilfælde udseende. En stor del af symbolerne er kun defineret i pakken amssymb hvilket der gøres opmærksom på i oversigterne.

Navngivne funktioner

Nedennævnte funktioner hedder i LaTeX-jargonen 'log-like functions'. – Funktionerne i den venstre spalte har den ekstra egenskab at eventuelle fodtegn (toptegn) kommer til at stå under (over) symbolet i en fremhævet formel.

lim	`\lim`	log	`\log`	exp	`\exp`
lim sup	`\limsup`	lg	`\lg`	ln	`\ln`
lim inf	`\liminf`	sin	`\sin`	cos	`\cos`
min	`\min`	arcsin	`\arcsin`	arccos	`\arccos`
max	`\max`	sinh	`\sinh`	cosh	`\cosh`
inf	`\inf`	tan	`\tan`	cot	`\cot`
sup	`\sup`	tanh	`\tanh`	coth	`\coth`
det	`\det`	arctan	`\arctan`	arg	`\arg`
gcd	`\gcd`	sec	`\sec`	csc	`\csc`
		ker	`\ker`	dim	`\dim`
		hom	`\hom`	deg	`\deg`
		Re	`\Re`	Im	`\Im`

Symboler med variabel størrelse

Afhængigt af om de står i en fremhævet formel eller i en tekstformel ændrer disse symboler størrelse, og eventuelle fod- og toptegn får ændret placering.

$\sum$	`\sum`	$\prod$	`\prod`	$\int$	`\int`
$\biguplus$	`\biguplus`	$\coprod$	`\coprod`	$\oint$	`\oint`
$\bigoplus$	`\bigoplus`	$\bigotimes$	`\bigotimes`	$\bigodot$	`\bigodot`
$\bigcap$	`\bigcap`	$\bigcup$	`\bigcup`	$\bigsqcup$	`\bigsqcup`
$\bigwedge$	`\bigwedge`	$\bigvee$	`\bigvee`		

Parenteser og parentesagtige tegn

Nedenstående tegn der i LaTeX-jargonen hedder 'delimiters', kan bruges alene, men kan også indgå i `\left` `\right` konstruktioner, se side 45.

(	(	)	)	[	[	]	]
{	\lbrace	}	\rbrace	⟨	\langle	⟩	\rangle
\	\backslash	/	/	|	\\|	‖	\\\|
⌈	\lceil	⌉	\rceil	⌊	\lfloor	⌋	\rfloor
↕	\updownarrow	↑	\uparrow	↓	\downarrow		
⇕	\Updownarrow	⇑	\Uparrow	⇓	\Downarrow		

I stedet for kommandoerne `\lbrace` og `\rbrace` kan man bruge `\{` og `\}`.

Plus, minus osv.:

+	+	−	-
±	\pm	∓	\mp
×	\times	÷	\div
·	\cdot	/	/
⊙	\odot		
Findes kun i amssymb:			
⊞	\boxplus	⊟	\boxminus
⊠	\boxtimes	⊡	\boxdot
⊕	\oplus	⊖	\ominus
⊗	\otimes	⊘	\oslash
⋉	\ltimes	⋊	\rtimes
∔	\dotplus	∙	\centerdot
⋇	\divideontimes		

Lighed og ulighed

Lighedstegn o.lgn.:

=	=	≠	\neq	≡	\equiv
≐	\doteq				
∼	\sim	≃	\simeq	≈	\approx
∝	\propto	≍	\asymp	≅	\cong
Findes kun i amssymb:					
≗	\circeq	≜	\triangleq	≡	\eqcirc
≑	\doteqdot	≒	\risingdotseq	≒	\fallingdotseq
∝	\varpropto	≏	\bumpeq	≎	\Bumpeq
∽	\backsim	≌	\backsimeq	≂	\eqsim
∼	\thicksim	≈	\thickapprox	≊	\approxeq
≁	\nsim	≇	\ncong		

Ulighedstegn:

$<$	`<`	$>$	`>`
$\ll$	`\ll`	$\gg$	`\gg`
$\leq$	`\leq`	$\geq$	`\geq`
Findes kun i amssymb:			
$\leqslant$	`\lessdot`	$\geqslant$	`\gtrdot`
$\lll$	`\lll`	$\ggg$	`\ggg`
$\leqq$	`\leqq`	$\geqq$	`\geqq`
$\leqslant$	`\leqslant`	$\geqslant$	`\geqslant`
$\lesseqgtr$	`\lesseqgtr`	$\gtreqless$	`\gtreqless`
$\eqslantless$	`\eqslantless`	$\eqslantgtr$	`\eqslantgtr`
$\lesssim$	`\lesssim`	$\gtrsim$	`\gtrsim`
$\lesssim$	`\lesssim`	$\gtrsim$	`\gtrsim`
$\lvertneqq$	`\lvertneqq`	$\gvertneqq$	`\gvertneqq`
$\nleq$	`\nleq`	$\ngeq$	`\ngeq`
$\nless$	`\nless`	$\ngtr$	`\ngtr`
$\lneqq$	`\lneqq`	$\gneqq$	`\gneqq`
$\nleqslant$	`\nleqslant`	$\ngeqslant$	`\ngeqslant`
$\lneq$	`\lneq`	$\gneq$	`\gneq`
$\nleqq$	`\nleqq`	$\ngeqq$	`\ngeqq`
$\lnsim$	`\lnsim`	$\gnsim$	`\gnsim`
$\lnapprox$	`\lnapprox`	$\gnapprox$	`\gnapprox`
$\lessgtr$	`\lessgtr`	$\gtrless$	`\gtrless`
$\lesseqgtr$	`\lesseqgtr`	$\gtreqless$	`\gtreqless`

Krøllede ulighedstegn:

$\prec$	`\prec`	$\succ$	`\succ`
$\preceq$	`\preceq`	$\succeq$	`\succeq`
Findes kun i amssymb:			
$\curlyeqprec$	`\curlyeqprec`	$\curlyeqsucc$	`\curlyeqsucc`
$\preccurlyeq$	`\preccurlyeq`	$\succcurlyeq$	`\succcurlyeq`
$\precsim$	`\precsim`	$\succsim$	`\succsim`
$\precapprox$	`\precapprox`	$\succapprox$	`\succapprox`
$\nprec$	`\nprec`	$\nsucc$	`\nsucc`
$\npreceq$	`\npreceq`	$\nsucceq$	`\nsucceq`
$\precnsim$	`\precnsim`	$\succnsim$	`\succnsim`
$\precneqq$	`\precneqq`	$\succneqq$	`\succneqq`
$\precnapprox$	`\precnapprox`	$\succnapprox$	`\succnapprox`

Logik og mængdelære

Delmænge-symboler:

$\subset$	\subset	$\supset$	\supset
$\subseteq$	\subseteq	$\supseteq$	\supseteq
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq
Findes kun i amssymb:			
$\subseteqq$	\subseteqq	$\supseteqq$	\supseteqq
$\Subset$	\Subset	$\Supset$	\Supset
$\sqsubset$	\sqsubset	$\sqsupset$	\sqsupset
$\subsetneq$	\subsetneq	$\supsetneq$	\supsetneq
$\varsubsetneq$	\varsubsetneq	$\varsupsetneq$	\varsupsetneq
$\nsubseteq$	\nsubseteq	$\nsupseteq$	\nsupseteq
$\subsetneqq$	\subsetneqq	$\supsetneqq$	\supsetneqq
$\varsubsetneqq$	\varsubsetneqq	$\varsupsetneqq$	\varsupsetneqq
$\nsubseteqq$	\nsubseteqq	$\nsupseteqq$	\nsupseteqq

Andre mængdelære-symboler:

$\in$	\in	$\ni$	\ni
$\cap$	\cap	$\cup$	\cup
		$\uplus$	\uplus
$\setminus$	\setminus	$\emptyset$	\emptyset
$\sqcap$	\sqcap	$\sqcup$	\sqcup
Findes kun i amssymb:			
$\Cap$	\Cap	$\Cup$	\Cup
$\smallsetminus$	\smallsetminus	$\varnothing$	\varnothing
$\complement$	\complement		

Logik:

$\forall$	\forall	$\exists$	\exists
$\wedge$	\wedge	$\vee$	\vee
$\neg$	\neg	$\models$	\models
$\vdash$	\vdash	$\dashv$	\dashv
Findes kun i amssymb:			
		$\nexists$	\nexists
$\curlywedge$	\curlywedge	$\curlyvee$	\curlyvee
$\barwedge$	\barwedge	$\veebar$	\veebar
$\doublebarwedge$	\doublebarwedge	$\nvdash$	\nvdash
$\Vdash$	\Vdash	$\nVdash$	\nVdash
$\vDash$	\vDash	$\nvDash$	\nvDash
$\Vvdash$	\Vvdash	$\nVDash$	\nVDash
$\therefore$	\therefore	$\because$	\because

Geometri og geometriske symboler:

Cirkler og stjerner

$\circ$	\circ	$\bigcirc$	\bigcirc
$\bullet$	\bullet		
$\star$	\star	$*$	\ast
Findes kun i amssymb:			
$\circledcirc$	\circledcirc	$\circledast$	\circledast
$\bigstar$	\bigstar	$\circleddash$	\circleddash

Trekanter og firkanter:

$\triangle$	\triangle	$\diamond$	\diamond
$\triangleleft$	\triangleleft	$\triangleright$	\triangleright
$\bigtriangleup$	\bigtriangleup	$\bigtriangledown$	\bigtriangledown
Findes kun i amssymb:			
$\vartriangle$	\vartriangle	$\triangledown$	\triangledown
$\vartriangleleft$	\vartriangleleft	$\vartriangleright$	\vartriangleright
$\trianglelefteq$	\trianglelefteq	$\trianglerighteq$	\trianglerighteq
$\ntriangleleft$	\ntriangleleft	$\ntriangleright$	\ntriangleright
$\ntrianglelefteq$	\ntrianglelefteq	$\ntrianglerighteq$	\ntrianglerighteq
$\blacktriangle$	\blacktriangle	$\blacktriangledown$	\blacktriangledown
$\blacktriangleleft$	\blacktriangleleft	$\blacktriangleright$	\blacktriangleright
$\square$	\square	$\blacksquare$	\blacksquare
$\lozenge$	\lozenge	$\blacklozenge$	\blacklozenge

Geometri:

$\perp$	\perp	$\angle$	\angle
$\parallel$	\parallel	$\mid$	\mid
Findes kun i amssymb:			
$\measuredangle$	\measuredangle	$\sphericalangle$	\sphericalangle
$\between$	\between		
$\shortparallel$	\shortparallel	$\shortmid$	\shortmid
$\nparallel$	\nparallel	$\nmid$	\nmid
$\nshortparallel$	\nshortparallel	$\nshortmid$	\nshortmid
$\leftthreetimes$	\leftthreetimes	$\rightthreetimes$	\rightthreetimes

Pile:

		$\mapsto$	\mapsto
		$\longmapsto$	\longmapsto
$\leftarrow$	\leftarrow	$\rightarrow$	\rightarrow
$\Leftarrow$	\Leftarrow	$\Rightarrow$	\Rightarrow
$\longleftarrow$	\longleftarrow	$\longrightarrow$	\longrightarrow
$\Longleftarrow$	\Longleftarrow	$\Longrightarrow$	\Longrightarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow
$\rightleftharpoons$	\rightleftharpoons	$\Longleftrightarrow$	\iff
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow
$\uparrow$	\uparrow	$\downarrow$	\downarrow
$\Uparrow$	\Uparrow	$\Downarrow$	\Downarrow
$\updownarrow$	\updownarrow	$\Updownarrow$	\Updownarrow
$\nearrow$	\nearrow	$\searrow$	\searrow
$\swarrow$	\swarrow	$\nwarrow$	\nwarrow

Findes kun i amssymb:

$\leftarrowtail$	\leftarrowtail	$\rightarrowtail$	\rightarrowtail
$\twoheadleftarrow$	\twoheadleftarrow	$\twoheadrightarrow$	\twoheadrightarrow
$\leftrightsquigarrow$	\leftrightsquigarrow	$\rightsquigarrow$	\rightsquigarrow
$\looparrowleft$	\looparrowleft	$\looparrowright$	\looparrowright
$\Lsh$	\Lsh	$\Rsh$	\Rsh
$\curvearrowleft$	\curvearrowleft	$\curvearrowright$	\curvearrowright
$\circlearrowleft$	\circlearrowleft	$\circlearrowright$	\circlearrowright
$\leftleftarrows$	\leftleftarrows	$\rightrightarrows$	\rightrightarrows
$\upuparrows$	\upuparrows	$\downdownarrows$	\downdownarrows
$\leftrightarrows$	\leftrightarrows	$\rightleftarrows$	\rightleftarrows
$\leftrightharpoons$	\leftrightharpoons	$\rightleftharpoons$	\rightleftharpoons
$\upharpoonright$	\upharpoonright	$\downharpoonright$	\downharpoonright
$\upharpoonleft$	\upharpoonleft	$\downharpoonleft$	\downharpoonleft
$\nleftarrow$	\nleftarrow	$\nrightarrow$	\nrightarrow
$\nLeftarrow$	\nLeftarrow	$\nRightarrow$	\nRightarrow
$\nLeftrightarrow$	\nLeftrightarrow	$\nleftrightarrow$	\nleftrightarrow
$\Lleftarrow$	\Lleftarrow	$\Rrightarrow$	\Rrightarrow

Fremmede bogstaver som matematiske symboler

Hebraiske bogstaver:

$\aleph$	\aleph				

Findes kun i amssymb:

$\beth$	\beth	$\gimel$	\gimel	$\daleth$	\daleth

Græske bogstaver

Her er en liste over kommandoerne for de græske bogstaver til brug i matematik (hvis man skal skrive tekster på græsk så har LaTeX andre og bedre tilbud).

dansk navn	lille bogstav		stort bogstav	
alfa	α	`\alpha`	A	`\mathrm{A}`
beta	β	`\beta`	B	`\mathrm{B}`
gamma	γ	`\gamma`	Γ	`\Gamma`
delta	δ	`\delta`	Δ	`\Delta`
epsilon	ε	`\eps`	E	`\mathrm{E}`
eller	ϵ	`\epsilon`		
digamma			$\digamma$	`\digamma`
zeta	ζ	`\zeta`	Z	`\mathrm{Z}`
eta	η	`\eta`	H	`\mathrm{H}`
theta	θ	`\theta`	Θ	`\Theta`
eller	ϑ	`\vartheta`		
iota	ι	`\iota`	I	`\mathrm{I}`
kappa	κ	`\kappa`	K	`\mathrm{K}`
eller	$\varkappa$	`\varkappa`		
lambda	λ	`\lambda`	Λ	`\Lambda`
my	μ	`\mu`	M	`\mathrm{M}`
ny	ν	`\nu`	N	`\mathrm{N}`
ksi	ξ	`\xi`	Ξ	`\Xi`
omikron	o	o	O	`\mathrm{O}`
pi	π	`\pi`	Π	`\Pi`
eller	ϖ	`\varpi`		
ro	ρ	`\rho`	P	`\mathrm{P}`
eller	ϱ	`\varrho`		
sigma	σ	`\sigma`	Σ	`\Sigma`
tau	τ	`\tau`	T	`\mathrm{T}`
ypsilon	υ	`\upsilon`	Υ	`\Upsilon`
fi	ϕ	`\phi`	Φ	`\Phi`
eller	φ	`\varphi`		
khi	χ	`\chi`	X	`\mathrm{X}`
psi	ψ	`\psi`	Ψ	`\Psi`
omega	ω	`\omega`	Ω	`\Omega`

Bemærk: Kommandoen `\eps` findes kun i IMFUFA-LaTeX. I standard-LaTeX hedder den `\varepsilon`. Tegnene $\digamma$ og $\varkappa$ er defineret i amssymb-pakken.

Diverse:

Fra dagligdagen:

†	\dagger	‡	\ddagger
♠	\spadesuit	♣	\clubsuit
♡	\heartsuit	♢	\diamondsuit
♯	\sharp	♭	\flat
♮	\natural		
		♪	\musicalnote
€	\euro	‰	\promille
№	\numero	℃	\celsius

Kommandoerne under linjen findes kun i IMFUFA-LaTeX (og de kan i øvrigt også benyttes i tekstsammenhæng).

Blandet:

$\top$	\top	$\bot$	\bot	$\wr$	\wr
$\frown$	\frown	$\smile$	\smile	$\bowtie$	\bowtie
ℓ	\ell	$\wp$	\wp	$\hbar$	\hbar
Findes kun i amssymb:					
$\smallfrown$	\smallfrown	$\smallsmile$	\smallsmile	$\Bbbk$	\Bbbk
$/$	\diagup	$\backslash$	\diagdown	$\backprime$	\backprime
$\multimap$	\multimap	$\intercal$	\intercal	$\pitchfork$	\pitchfork
$\Finv$	\Finv	$\Game$	\Game	$\mho$	\mho
$\ni$	\backepsilon	$\eth$	\eth	$\hslash$	\hslash

Kapitel 6

Æstetiske betragtninger

Der er en lang række æstetiske regler indbygget i LaTeX, og for det meste står man sig ved at lade LaTeX administrere disse regler på egen hånd; i visse situationer er brugeren dog nødt til at have en finger med i spillet. Endvidere afhænger slutresultatets æstetiske kvaliteter naturligvis i nogen grad af hvordan brugeren skriver LaTeX-koden. Endelig er der visse parametre som brugeren normalt *ikke* bør ændre.

Læselighed

En af grundene til at skrive noget er at man har en forventning om at der vil være nogen der vil læse det man skriver, og derfor bør tekstens udseende tilrettelægges på en sådan måde at den bliver let at læse – det samme gælder naturligvis indholdet, men det er en anden historie. Der er mange parametre der spiller ind på læseligheden: bogstaverne må hverken være for store eller for små, linjerne hverken for korte eller for lange, linjeafstanden hverken for stor eller for lille, og skriften skal være klar og tydelig. Forskere har undersøgt hvad der medvirker til en optimal læsesituation (se for eksempel Ellegaard Frederiksen (1981) og litteraturhenvisninger deri). Hvad angår LaTeX-tekster er budskabet:

- Linjelængden skal ikke være større end den bliver med IMFUFA-LaTeXs dokumentklasser[1], snarere mindre.
- Skriftgraden skal være 10 punkt til normaltlæsende voksne; hvis der optræder matematiske symboler med top- og fodtegn i mange niveauer, kan det måske være nødvendigt at benytte 11 eller 12 punkt.
- Linjeafstanden må ikke være for lille. Standard LaTeX bruger 12 punkts skydning af 10 punkts skriften, og det er lovlig lidt. IMFUFA-LaTeXs dokumentklasser accepterer en option `10pt13` der giver en 10 punkts skrift med 13 punkts skydning; resultatet heraf ses i nærværende hæfte.
- Læseligheden afhænger i høj grad af valget af skrift. Man bør næppe skrive længere afsnit i blokskrift. NÆPPE HELLER I KAPITÆLER.

[1] som kalder Standard LaTeX med standard-option'en `a4paper`.

Den tidligere (side 29) anvendte Musil-tekstprøve sat på fire forskellige måder.

Egentlig var det kun matematikerne selv og deres disciple, naturforskerne, der levede i uvidenhed om disse farer, de fornemmede lige så lidt af alt dette i deres sjæl som cykelryttere, der flittigt tramper løs og ikke lægger mærke til andet i verden end den forankørendes baghjul. Om Ulrich kunne man derimod med sikkerhed sige det ene, at han elskede matematikken på grund af de mennesker, der ikke kunne udstå den. Han var mindre videnskabeligt end menneskeligt forelsket i videnskaben. Han så, at den i alle spørgsmål, hvor den betragter sig som kompetent, tænker anderledes end almindelige mennesker. Hvis man i stedet for videnskabelige anskuelser satte livsanskuelse, i stedet for hypotese forsøg og i stedet for sandhed handling, var der ingen betydelig naturforsker eller matematiker, hvis livsværk ikke langt overgik historiens største bedrifter.

Figur 6.1 Sat med `\fussy`

Egentlig var det kun matematikerne selv og deres disciple, naturforskerne, der levede i uvidenhed om disse farer, de fornemmede lige så lidt af alt dette i deres sjæl som cykelryttere, der flittigt tramper løs og ikke lægger mærke til andet i verden end den forankørendes baghjul. Om Ulrich kunne man derimod med sikkerhed sige det ene, at han elskede matematikken på grund af de mennesker, der ikke kunne udstå den. Han var mindre videnskabeligt end menneskeligt forelsket i videnskaben. Han så, at den i alle spørgsmål, hvor den betragter sig som kompetent, tænker anderledes end almindelige mennesker. Hvis man i stedet for videnskabelige anskuelser satte livsanskuelse, i stedet for hypotese forsøg og i stedet for sandhed handling, var der ingen betydelig naturforsker eller matematiker, hvis livsværk ikke langt overgik historiens største bedrifter.

Figur 6.2 Sat med `\sloppy`

Egentlig var det kun matematikerne selv og deres disciple, naturforskerne, der levede i uvidenhed om disse farer, de fornemmede lige så lidt af alt dette i deres sjæl som cykelryttere, der flittigt tramper løs og ikke lægger mærke til andet i verden end den forankørendes baghjul. Om Ulrich kunne man derimod med sikkerhed sige det ene, at han elskede matematikken på grund af de mennesker, der ikke kunne udstå den. Han var mindre videnskabeligt end menneskeligt forelsket i videnskaben. Han så, at den i alle spørgsmål, hvor den betragter sig som kompetent, tænker anderledes end almindelige mennesker. Hvis man i stedet for videnskabelige anskuelser satte livsanskuelse, i stedet for hypotese forsøg og i stedet for sandhed handling, var der ingen betydelig naturforsker eller matematiker, hvis livsværk ikke langt overgik historiens største bedrifter.

Figur 6.3 Sat med `\raggedright`

Egentlig var det kun matematikerne selv og deres disciple, naturforskerne, der levede i uvidenhed om disse farer, de fornemmede lige så lidt af alt dette i deres sjæl som cykelryttere, der flittigt tramper løs og ikke lægger mærke til andet i verden end den forankørendes baghjul. Om Ulrich kunne man derimod med sikkerhed sige det ene, at han elskede matematikken på grund af de mennesker, der ikke kunne udstå den. Han var mindre videnskabeligt end menneskeligt forelsket i videnskaben. Han så, at den i alle spørgsmål, hvor den betragter sig som kompetent, tænker anderledes end almindelige mennesker. Hvis man i stedet for videnskabelige anskuelser satte livsanskuelse, i stedet for hypotese forsøg og i stedet for sandhed handling, var der ingen betydelig naturforsker eller matematiker, hvis livsværk ikke langt overgik historiens største bedrifter.

Figur 6.4 Sat med `\RaggedRight`

- I en skønlitterær tekst skal ordmellemrummene have stort set samme bredde. Det skal de også i en matematisk tekst, men der kan det være sværere at opnå eftersom formler ikke bør deles ved linjeskift.

  Angående ordmellemrum og orddeling har LaTeX to grundindstillinger der vælges med kommandoerne `\fussy` og `\sloppy`:

  - Med `\fussy` får man ordmellemrum af samme størrelse. Prisen herfor er at linjer med meget lange tekstdele kan gå ud over højrekanten af teksten, og man får advarslen `Overfull \hbox`. Se Figur 6.1.
  - Med `\sloppy` vil linjerne næsten aldrig gå ud over højrekanten. Til gengæld kan der komme meget store mellemrum mellem ordene, og man får advarslen `Underfull \hbox`. Se Figur 6.2.

- Normalt bliver teksten sat med *fast bagkant* (lige højremargin), men det

kan også lade sig gøre at få *løs bagkant*, sådan som det efterhånden ses i de bedre aviser og i en del bøger. I Figur 6.3 og Figur 6.4 er den samme tekst sat med løs bagkant på to forskellige måder. I Figur 6.3 er benyttet kommandoen `\raggedright` der bevirker at når et ord er for langt til at stå på linjen, så bliver det rykket ned på næste linje, dvs. ingen ord bliver delt. I Figur 6.4 er benyttet IMFUFA-LaTeX-kommandoen `\RaggedRight` der tillader en vis variation i linjelængden, men som også tillader orddeling.

- Nyt afsnit markeres almindeligvis ved linjeskift efterfulgt af en vis mindre indrykning, jf. side 8. Denne metode som er LaTeXs standardmetode, er benyttet i nærværende publikation

  Alternativt kan man markere nyt afsnit ved linjeskift med ekstra nedrykning efterfulgt af ingen indrykning; det gør LaTeX i opremsninger som for eksempel den du læser i øjeblikket.

  I IMFUFA-LaTeX kan man vælge mellem de to metoder, idet man kan angive `imfufapar` som option til dokumentklassen for at få den alternative.

Man kan naturligvis diskutere om man *altid* skal stræbe efter at gøre tekstbilledet så læse-bekvemt som muligt; tekster der kræver eftertanke og fordybelse skal måske netop udformes på en måde der *nedsætter* læsehastigheden ...

Streger og mellemrum

Der er talrige småting der kan genere og irritere den pedantiske læser – også selv om vi ser helt bort fra tekstens *indhold*.

Bindestreger, tankestreger og minusser er både syntaktisk og i LaTeX-forstand tre helt forskellige ting.

Symbolet for tankestreger er *to* streger – og der skal mellemrum før og efter.	`Symbolet for tankestreger er \emph{to} streger -- og der skal mellemrum før og efter.`

EKSEMPEL 6.1

Bindestreg skrives med en enkelt streg: 1990-95, kl.8-16.	`Bindestreg skrives med en enkelt streg: 1990-95, kl.8-16.`

EKSEMPEL 6.2

I visse sammensatte ord (f.eks. computer-specialist) kan man indsætte en bindestreg.	`I visse sammensatte ord (f.eks. computer-specialist) kan man indsætte en bindestreg.`

EKSEMPEL 6.3

LaTeX deler normalt ikke ord der er sat sammen med en bindestreg. Men hvis man i stedet for bindestregen skriver `"-`, er LaTeX villig til at dele de to orddele:

EKSEMPEL 6.4

I visse sammensatte ord (f.eks. computer-specialist) kan man indsætte en bindestreg.	`I visse sammensatte ord (f.eks. computer"-specialist) kan man indsætte en bindestreg.`

Denne facilitet skyldes den danske del af babel-pakken. IMFUFA-LaTeX definerer i øvrigt også en kommando "/ der virker på lignende måde, blot skriver den en / i stedet for en -.

Minustegn er atter noget andet; de hører ikke hjemme i tekst- men i matematiksammenhæng:

EKSEMPEL 6.5

Det er typografisk og æstetisk ukorrekt at skrive 5-7=-2. I stedet skal man skrive $5 - 7 = -2$.	`Det er typografisk og æstetisk ukorrekt at skrive 5-7=-2. I stedet skal man skrive $5-7=-2$.`

Den skarpøjede læser vil i Eksempel 6.5 kunne se at LaTeX i matematiksammenhæng skelner mellem den dyadiske operator minus og fortegnet minus således at der kommer større mellemrum omkring den første.

LaTeX kan selv sørge for at ordmellemrummene får en passende størrelse, men brugeren skal selv markere *hvor* der skal være mellemrum og hvor ikke. Det indebærer at den bruger der vil følge almindelig dansk skik, selv skal skrive mellemrumstegn *efter* (men ikke foran) tegn som punktum, semikolon, kolon, komma, udråbstegn og spørgsmålstegn. Det har samme effekt om man skriver et eller mange mellemrumstegn; i de fleste situationer som den almindelige bruger kommer ud for, har et enkelt linjeskift samme betydning som et mellemrumstegn.

Der er forskellige sammenhænge hvor man *ikke* må skrive mellemrum, for eksempel først i argumentet til en 'sectioning command':

EKSEMPEL 6.6

Overskriften bør flugte med venstrekanten:	`Overskriften bør flugte med venstrekanten:`
Forkert	`\subsection{ Forkert}`
Rigtigt	`\subsection{Rigtigt}`

Et mellemrum der ikke må erstattes af linjeskift, skrives som en tilde (~):

EKSEMPEL 6.7

Som Valdemar Atterdag sagde: I morgen er der atter en dag.	`Som Valdemar Atterdag sagde: I~morgen er der atter en dag.`

Kapitel 7

Om at inkludere grafik

LaTeX er beregnet til tekst og matematiske formler, ikke til billeder. Hvis man alligevel vil have billeder eller grafiske fremstillinger ind i sin tekst, er der to principielt forskellige løsninger: enten kan man fremstille billedet ved hjælp af LaTeX-kommandoer (se `picture`), eller også kan man lade LaTeX afsætte noget plads hvor man senere kan indsætte et billede som er fremstillet på anden vis.

Operationerne *afsæt noget plads* og *indsæt et billede* kan foregå mere eller mindre sofistikeret. Den mest jordbundne fremgangsmåde er at have billedet på et stykke papir som man klistrer ind på en dertil afsat plads i den færdige tekst; i den forbindelse kan man have fornøjelse af kommandoen `\vspace` der indsætter vertikalt (lodret) mellemrum af en given størrelse, f.eks. `\vspace{5cm}`.

Ofte foreligger billedet i form af en fil fremstillet af et eller andet andet program; så kan man i præamblen inkludere pakken graphicx[1] (Carlisle & Rahtz, 1999) der definerer kommandoen `\includegraphics` som kan gøre to ting: Når LaTeX sætter teksten, henter kommandoen information om billedets størrelse ved at læse i billedfilen (i visse tilfælde er brugeren dog nødt til selv at angive størrelsen), og når dokumentet bliver udskrevet, bliver billedet indsat på det ønskede sted (forudsat at den benyttede printer faktisk er i stand til at udskrive grafik af den anvendte type). Den type billedfiler som alt i alt er lettest at håndtere, er de såkaldte Encapsulated PostScript filer (eps-filer).

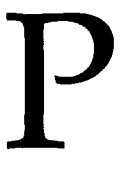
EPS-grafik

Det er ganske let at indføje grafik i form af en eps-fil i et LaTeX-dokument. Man skal blot bede om at benytte graphicx-pakken og så anvende kommandoen `\includegraphics`. – Der er dog den besværlighed at ikke alle viewere kan *vise* grafikken, og man er nødt til at skrive ud på en PostScript-printer. I den forbindelse er der et tip: Hvis man i stedet for `\usepackage{graphicx}` skriver `\usepackage[draft]{graphicx}`, så erstattes grafikken (som man ikke kan se med vieweren) med en box der viser hvor meget plads billedet vil optage, og inde i boxen skrives navnet på grafik-filen.

[1] Denne pakke er en standard LaTeX-pakke der definerer en autoriseret generel metode til inklusion af grafik.

Til demonstration af grafikhåndtering vil vi benytte et billede der »forestiller« Palatino-skriftens P; billedet findes i en fil ved navn `p.eps`.

EKSEMPEL 7.1

Her er et stort P:  Er det pænt?

```
Her er et stort P:
\includegraphics{p.eps}
Er det pænt?
```

Figuren placeres så dens nederste kant står på linje med den almindelige tekst. Det kan man ændre på (se Eksempel 7.5), men først vil vi af pladshensyn give figuren en anden størrelse. Man kan *skalere* figuren op eller ned:

EKSEMPEL 7.2

Skaleret til halv størrelse: 

```
Skaleret til halv størrelse:
\includegraphics[scale=0.5]{p.eps}
```

og man kan give figuren *en bestemt størrelse*:

EKSEMPEL 7.3

Med en given størrelse: 

```
Med en given størrelse:
\includegraphics[width=1cm]{p.eps}
```

Som det ses, regulerer `width` figurens udstrækning i begge dimensioner. Man kan sætte `width` til en vilkårlig af LaTeX kendt længde, også for eksempel et tal gange `\linewidth` eller et tal gange `\textwidth`.

Som ethvert andet LaTeX-objekt kan resultatet af `\includegraphics` indrammes med kommandoen `\fbox` (se også side 91):

EKSEMPEL 7.4

Indrammet P:  .

```
Indrammet P:
\fbox{\includegraphics[scale=0.5]
  {p.eps}}.
```

Som ethvert andet LaTeX-objekt kan resultatet af `\includegraphics` rykkes op eller ned med kommandoen `\raisebox` (se også side 92):

EKSEMPEL 7.5

Rykket 3 mm ned:  .

```
Rykket 3 mm ned:
\raisebox{-3mm}{\includegraphics
  [scale=0.5]{p.eps}}.
```

Hvis billedet skal placeres centreret på en linje for sig selv, er det lettest at gøre det med `\[ \]`:

Her er figuren centreret på en linje for sig selv:

Her fortsætter teksten.

```
Her er figuren centreret på
en linje for sig selv:
\[
\includegraphics[scale=0.5]{p.eps}
\]
Her fortsætter teksten.
```

EKSEMPEL 7.6

Man kan rotere figuren et ønsket antal grader:

Skalér til halv størrelse og rotér 90 grader i negativ omløbsretning:

```
Skalér til halv størrelse
og rotér 90 grader i negativ
omløbsretning: \includegraphics
[scale=0.5,angle=-90]{p.eps}
```

EKSEMPEL 7.7

Fikspunktet for rotationen er normalt nederste venstre hjørne. Det kan ændres med `origo=`⟨*specifikator*⟩, hvor ⟨*specifikator*⟩ kan være et eller to af `c l r t b B` (for hhv. center, left, right, top, bottom, Baseline):

Rotation om centrum:  .

```
Rotation om centrum:
\includegraphics[scale=0.5,
  origin=c,angle=-90]{p.eps}.
```

EKSEMPEL 7.8

Bemærkning: I listen over manipulationer (rotationer, størrelsesændringer o.lgn.) der skal foretages på grafikfilen, spiller *rækkefølgen* en rolle (fordi operationerne udføres i den rækkefølge de er skrevet), eksempelvis giver `width=3cm,rotate=45` ikke det samme som `rotate=45,width=3cm`.

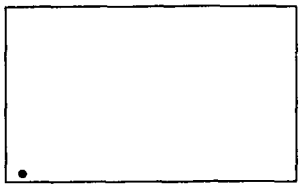
Grafik med `picture`

LaTeX har en `picture`-konstruktion der kan bruges til simple tegninger. – Den syntaks der gælder inden for en `picture`-konstruktion, stemmer ikke helt overens med den almindelige LaTeX-syntaks, så der er rige muligheder for at begå fejl.

Vi vil nu lave et billed-objekt som er 34 mm bredt og 21 mm højt, og hvor der er placeret en sort klat i det punkt som har koordinaterne $(0,0)$ i det lokale koordinatsystem; der et sat en ramme uden om billed-objektet (ved hjælp af `\fbox`) så man kan se hvor meget plads LaTeX har afsat til det:

EKSEMPEL 7.9

Se her:



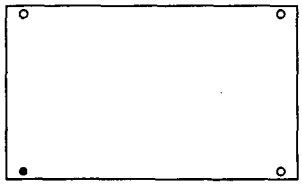
```
Se her:
\setlength{\unitlength}{1mm}
\fbox{
\begin{picture}(34,21)
  \put(0,0){\circle*{1}}
\end{picture}
}
```

Der sker følgende: Før man begynder på billedet, fastsættes den længdeenhed (`\unitlength`) der skal benyttes; her er den sat til 1 mm, men man kan sætte den til en hvilken som helst længde (standardværdien er 1 pt). Derefter oprettes et billed-objekt hvis bredde og højde er hhv. 34 og 21 længdeenheder. Endelig anbringes en udfyldt cirkel med radius 1 i punktet $(0,0)$. Med hensyn til syntaksen skal man bemærke at der er en del størrelser der skal stå i *runde* parenteser.

Kommandoen `\put` placerer et billed-objekt på et givet sted i det lokale koordinatsystem. Lad os placere ikke-udfyldte cirkler i de tre andre hjørner:

EKSEMPEL 7.10

Se her:

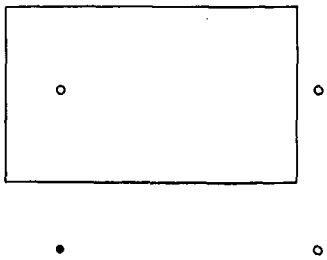


```
Se her:
\setlength{\unitlength}{1mm}
\fbox{
\begin{picture}(34,21)
  \put(0,0){\circle*{1}}
  \put(0,21){\circle{1}}
  \put(34,0){\circle{1}}
  \put(34,21){\circle{1}}
\end{picture}
}
```

Efter `\begin{picture}` *skal* der stå et talpar, men der kan godt stå to:

EKSEMPEL 7.11

Se her:



```
Se her:
\setlength{\unitlength}{1mm}
\fbox{
\begin{picture}(34,21)(-5,10)
  \put(0,0){\circle*{1}}
  \put(0,21){\circle{1}}
  \put(34,0){\circle{1}}
  \put(34,21){\circle{1}}
\end{picture}
}
```

Det andet talpar forskyder billedets indhold; talparret (x,y) bevirker en forskydning på $(-x,-y)$. Bemærk at det netop kun er billedets *indhold* der forskydes,

selve billed-objektet ændrer ikke placering eller størrelse, hvilket ses af indramningerne. – I de følgende eksempler undlader vi indramningen.

Man kan anbringe tekst i billedet; for at vise hvordan teksten placeres, anbringer vi en udfyldt cirkel samme sted:

Se her:

IMFUFA
IMFUFA
IMFUFA
IMFUFA

```
Se her:
\setlength{\unitlength}{1mm}
\begin{picture}(34,21)
\put(17,18){\circle*{1}}
\put(17,18){IMFUFA}
\put(17,13){\circle*{1}}
\put(17,13){\makebox(0,0){IMFUFA}}
\put(17,8){\circle*{1}}
\put(17,8){\makebox(0,0)
   [r]{IMFUFA}}
\put(17,3){\circle*{1}}
\put(17,3){\makebox(0,0)
   [lb]{IMFUFA}}
\end{picture}
```

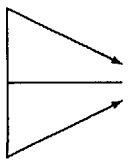
EKSEMPEL 7.12

Det talpar der står efter `\makebox`, er størrelsen af den fremstillede box. Boxen placeres med nederste venstre hjørne i det punkt som angives i `\put`, med mindre der er angivet en plads-specifikator i `[ ]`.

Foruden `\makebox` findes også kommandoerne `\framebox` og `\dashbox` der virker på tilsvarende måde, blot omgiver de deres indhold med en fuldt optrukket hhv. stiplet ramme.

Man kan tegne linjestykker og vektorer (linjestykker med en pil i den ene ende) som har en rational hældningskoefficient p/q eller som er lodrette. Syntaksen er `\line(p,q){`⟨*længde*⟩`}` eller `\vector(p,q){`⟨*længde*⟩`}` hvor ⟨*længde*⟩ er længden af linjens projektion på y-aksen hvis $p = 0$ og linjen dermed er lodret, eller længden af projektionen på x-aksen. I alle tilfælde skal ⟨*længde*⟩ være større eller lig 0. Linjestykket afsættes ud fra det punkt der specificeres i `\put`.

Se her:



```
Se her:
\setlength{\unitlength}{1mm}
\begin{picture}(34,21)
  \put(10, 0){\vector(2,1){15}}
  \put(10,10){\line(1,0){15}}
  \put(10,20){\vector(2,-1){15}}
  \put(10, 0){\line(0,1){20}}
\end{picture}
```

EKSEMPEL 7.13

På grund af interne begrænsninger i LaTeX kan man kun benytte temmelig små værdier af p og q. – Hvis man benytter emTeX, er der dog mulighed for at tegne linjer med stort set vilkårlig hældningskoefficient; man skal da bruge pakken emlines2 hvori der defineres kommandoen `\emline` der kan tegne en linje fra et punkt til et andet:

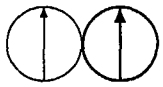
EKSEMPEL 7.14

Se her: 

```
Se her:
\setlength{\unitlength}{1mm}
\begin{picture}(34,21)
  \put(10, 0){\circle*{1}}
  \put(20,17.5){\circle{1}}
  \emline{10}{0}{}{20}{17.5}{}
\end{picture}
```

Linjer der tegnes med standard LaTeX-kommandoer (ikke `\emline`), kan gøres tykkere med kommandoen `\thicklines`; kommandoen `\thinlines` giver normal tykkelse.

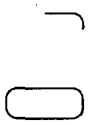
EKSEMPEL 7.15

Se her: 

```
Se her:
\setlength{\unitlength}{1mm}
\begin{picture}(34,21)
  \put(10,10){\circle{10}}
  \put(10, 5){\vector(0,1){10}}
  \thicklines
  \put(20,10){\circle{10}}
  \put(20, 5){\vector(0,1){10}}
\end{picture}
```

Foruden cirkler kan man tegne ovaler:

EKSEMPEL 7.16

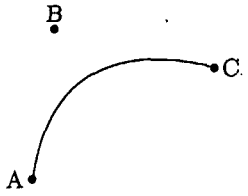
Se her: 

```
Se her:
\setlength{\unitlength}{1mm}
\begin{picture}(34,21)
  \put(15,5){\oval(10,4)}
  \put(15,15){\oval(10,4)[tr]}
\end{picture}
```

Ligesom der kun er et begrænset udvalg af hældningskoefficienter for linjer tegnet med `\line` og `\vector`, er der også kun et begrænset antal diametre for cirkler og ovaler.

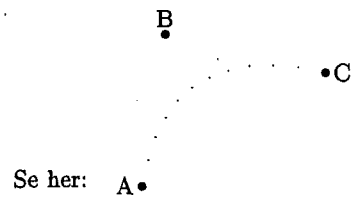
Kommandoen `\qbezier` tegner en kvadratisk Bézier-kurve; her forbindes punkterne A og C med en Bézier-kurve med B som det tredje Bézier-punkt:

EKSEMPEL 7.17

Se her: 


```
Se her:
\setlength{\unitlength}{1mm}
\begin{picture}(34,21)
  \put(5,0){\circle*{1}}
  \put(4,0){\makebox(0,0)[r]{A}}
  \put(29,15){\circle*{1}}
  \put(30,15){\makebox(0,0)[l]{C}}
  \put(8,20){\circle*{1}}
  \put(8,21){\makebox(0,0)[b]{B}}
  \qbezier(5,0)(8,20)(29,15)
```

Den tegnede Bézier-kurve består »i virkeligheden« af tætliggende punkter; man kan selv bestemme hvor mange punkter den skal bestå af, her består den af 11 punkter:




```
Se her:
\setlength{\unitlength}{1mm}
\begin{picture}(34,21)
  \put(5,0){\circle*{1}}
  \put(4,0){\makebox(0,0)[r]{A}}
  \put(29,15){\circle*{1}}
  \put(30,15){\makebox(0,0)[l]{C}}
  \put(8,20){\circle*{1}}
  \put(8,21){\makebox(0,0)[b]{B}}
  \qbezier[11](5,0)(8,20)(29,15)
\end{picture}
```

EKSEMPEL 7.18

Bézier-funktionen er en del af Standard LaTeX og kræver således ikke at man benytter specielle pakker.

Som det fremgår af det foregående, kan man faktisk fremstille ret avancerede tegninger, men det kræver tid og tålmodighed. En del matematik- og statistikprogrammer kan eksportere tegninger, f.eks. af funktionsgrafer eller af datapunkter, i form af (en fil der indeholder) et LaTeX-`picture`.

Kapitel 8

Videregående emner

Stikord og stikordsregister

I visse typer publikationer er det en stor hjælp med et fornuftigt stikordsregister. Man kan sagtens lave stikordsregistre til LaTeX-dokumenter, men det er en proces der består af flere trin hvoraf nogle er lidt omstændelige:

1. Man skal beslutte sig for hvilke ord i teksten der skal tilknyttes et stikord. Det gøres af forfatteren eller en anden person.

2. I `tex`-filerne skal man indsætte `\index`-kommandoer i overensstemmelse med vedtagelsen i forrige punkt. Kommandoen `\index` omtales nærmere i det følgende.

3. Man skal instruere LaTeX om at der skal laves et stikordsregister. Det foregår ved at man skriver `\makeindex` i præamblen. Virkningen heraf er at LaTeX fremstiller en hjælpefil (med efternavn `idx`) hvor der for hver `\index`-kommando kommer en linje med det pågældende stikord og det tilhørende sidenummer.

4. Stikordslisten skal sorteres så stikordene kommer i alfabetisk orden og uden dubletter. Det gøres af et særligt stikordsbehandlingsprogram, f.eks. det klassiske MakeIndex der findes i mange LaTeX-installationer, eller det nyere xindy; under DOS kan man desuden bruge MkInd.[1] Under alle omstændigheder bliver resultatet en såkaldt indexfil (med efternavnet `ind`).

5. Indexfilen skal inkluderes i dokumentet.

[1] MkInd er stikordsregisterprogram som er skrevet af Jørgen Larsen og som sorterer efter danske alfabetiseringsregler. Man kan sige at MkInd er en delmængde af MakeIndex i den forstand at `\index`-kommandoer skrevet med henblik på det første også forstås (og – med en enkelt undtagelse – forstås på samme måde) af det andet, forudsat at man i MakeIndex omdefinerer specialtegnene som beskrevet i Afsnit 12.4.3 i Goossens et al. (1994). (Dette valg af specialtegn synes velegnet til nordeuropæiske sprog som tysk, dansk og hollandsk, der også i TeX-sammenhæng har store ligheder.) Undtagelsen er at MakeIndex i modsætning til MkInd opfatter *alle* blanke tegn som betydende. Derudover sorterer MakeIndex, som man måske kan gætte, efter engelske alfabetiseringsregler (eller tyske hvis man beder om det).

(Hvis primærfilen hedder `projekt.tex`, så kommer indexfilen til at hedde `projekt.ind`, og man skal altså skrive `\input{projekt.ind}` i slutningen af primærfilen.)

Herefter skal man køre dokumentet igennem LaTeX et par gange og kalde stikordsbehandlingsprogrammet efter hver gang således at stikordsregisteret kommer med, også i indholdsfortegnelsen, samt læse korrektur en ekstra gang.

Kommandoen `\index`

Den simpleste form for kald af `\index` er `\index{`⟨*stikord*⟩`}`, for eksempel

```
Vi er underkastet åndsfyrsternes\index{åndsfyrste} herredømme.
```

Man bør så vidt muligt undgå mellemrum mellem `\index`-kommandoen og det ord den henviser til; ethvert mellemrum rummer jo en mulighed for et sideskift, og i så fald kan der blive henvist til en forkert side.

Som ⟨*stikord*⟩ kan man skrive næsten hvad som helst, dog skal man være opmærksom på at der er visse tegn der fungerer som specialtegn i forhold til stikordsbehandlingsprogrammet. Hvis man bruger MkInd (eller MakeIndex som beskrevet i fodnote 1) er de fire tegn `!` `>` `=` `|` specialtegn med følgende betydning:

- Man bruger tegnet `=` hvis stikordet skal alfabetiseres på en anden måde end det skrives. Syntaksen er `\index{`⟨*nøgle*⟩ `=` ⟨*stikord*⟩`}` .

  Hvis man eksempelvis skriver

  ```
  \index{Ålborg = Aalborg}
  ```

  så kommer stikordet »Aalborg« til at optræde i listen under Å (og i øvrigt umiddelbart efter stikordet »Ålborg«, hvis der skulle være et sådant).

  Her er to andre eksempler; det første bevirker at stikordet `\beta` kommer til at stå som om det var stavet beta, det andet bevirker at stikordet β kommer til at stå som om det var et b:

  ```
  \index{beta = \texttt{\textbackslash beta}}
  \index{b = $\beta$}
  ```

- Man bruger tegnet `>` for at lave underpunkter og underunderpunkter til et stikord. Syntaksen er `\index{`⟨*punkt*⟩ `>` ⟨*underpunkt*⟩`}` hhv. `\index{`⟨*punkt*⟩ `>` ⟨*underpunkt*⟩ `>` ⟨*underunderpunkt*⟩`}` .

  Eksempel:

  ```
  \index{åndsfyrste > mindre}
  ```

  Som ⟨*punkt*⟩ eller ⟨*underpunkt*⟩ kan man godt skrive noget af formen ⟨*nøgle*⟩=⟨*stikord*⟩.

- Man kan bruge tegnet `|` for at få sidetallet hørende til en bestemt forekomst af `\index` skrevet med en særlig skrift (for eksempel for at markere at emnet får en særlig udførlig behandling på denne side).

  Eksempel: hvis man skriver

  ```
  \index{åndsfyrste |textbf }
  ```

  så bliver sidetallet skrevet med skriften svarende til `\textbf`.

  Hvis man skriver

  ```
  \index{åndsfyrste |see intellektuelt udfordret person }
  ```

  så kommer der i den endelige stikordsliste ved »åndsfyrste« en henvisning til »intellektuelt udfordret person«.

- Man bruger tegnet `!` for at ophæve et af specialtegnenes virkning.

  Eksempel: Vi vil gerne have formlen $E = mc^2$ til at optræde i stikordsregisteret under E. Da = er et specialtegn, kan vi ikke skrive `\index{$E=mc^2$}`, vi er nødt til at normalisere lighedstegnet ved at sætte et `!` foran: `\index{$E!=mc^2$}`. Dette `\index`-kald resulterer i stikordet $E = mc^2$, det kommer bare ikke til at stå under E, men under $. Derfor må vi bruge = for at fortælle hvordan det skal alfabetiseres; den rigtige løsning er

  ```
  \index{E!=mc^2 = $E!=mc^2$}
  ```

Litteraturliste og -henvisninger

Visse typer skriftlige arbejder skal indeholde en litteraturliste, og der skal så i selve teksten være henvisninger til værker der er opført i litteraturlisten. Det giver anledning til en del spørgsmål (og svar som uddybes i det følgende):

1. Hvordan organiserer man en litteraturdatabase på en hensigtsmæssig måde?

   Som en BibTeX-database.

2. Hvordan fremstiller og vedligeholder man en BibTeX-database?

   Enten med »håndkraft« (databasen er en almindelig tekstfil) eller med et BibTeX-databaseprogram; under DOS eller MS-Windows kan man f.eks. bruge BibDB.

3. Hvordan instruerer man LaTeX om at benytte en bestemt BibTeX-base?

   Med kommandoen `\bibliography`.

4. Hvordan henviser man til noget i litteraturlisten?

   Med kommandoen `\cite`.

```
@book{companion,
   author = {Michel Goossens and Frank Mittelbach and Alexander Samarin},
   title = {The {\LaTeX} Companion},
   year = 1994,
   publisher = {Addison-Wesley},
   address = {Reading, Massachusetts}
}

@BOOK{Ellegaard:81,
   author = {Erik {Ellegaard Frederiksen}},
   title = {Bogens Funktion og {\AE}stetik},
   year = 1981,
   publisher = {Christian Ejlers' Forlag},
   address = {K{\o}benhavn}
}
```

Figur 8.1 Et uddrag af BibTeX-basen der er benyttet til nærværende skrift.

5. Hvordan bestemmer man hvordan henvisningen skal udformes, f.eks. om det skal være et tal i kantede parenteser, eller efter modellen ⟨*forfatter*⟩ (⟨*årstal*⟩) (f.eks. Dahl (1944))?

   LaTeX har en standardmetode, nemlig et tal i kantede parenteser. Dette er grimt, og man bør derfor benytte en velvalgt pakke, for eksempel natbib, *og* en velvalgt bibliografistil for at få pænere henvisninger.

6. Hvordan bestemmer man hvordan litteraturlisten skal se ud?

   Med LaTeX-kommandoen `\bibliographystyle`.

BibTeX-databaser

Litteraturdatabaser organiseres hensigtsmæssigt som som BibTeX-databaser. BibTeX-databaser kan være meget store (man kunne f.eks. have en base der omfattede »alt« om matematiske modeller), LaTeX kan selv finde ud af at uddrage de nødvendige oplysninger til en litteraturliste til en konkret tekst.

En BibTeX-database er en tekstfil (med efternavn `bib`) hvor hver enkelt bog, artikel osv. er opført som en post efter et særligt system. I Figur 8.1 ses to sådanne poster (svarende til værkerne Goossens et al. (1994) og Ellegaard Frederiksen (1981)). For hver post angives først hvad *type* publikation der er tale om, jf. Tabel 8.1; dernæst følger den *nøgle* man skal bruge når man henviser til publikationen, og endelig felter med de tilgængelige *bibliografiske oplysninger*, jf. Tabel 8.2. Som det ses, er det næsten normal LaTeX-tekst der står, men bemærk at det ikke er europæisk, men amerikansk standard; det er således ikke tilladt at skrive f.eks. æ ø å.

Det tidligere nævnte program BibDB kan benyttes til at administrere en BibTeX-database; programmet kender de forskellige publikationstyper og véd hvilke oplysninger hver af dem kræver. Man må gerne bruge æ ø å i BibDB, programmet laver dem selv om til de fornødne LaTeX-koder.

Tabel 8.1 Almindeligt kendte publikationstyper

`article`
`book`
`booklet`
`inbook`
`incolletion`
`proceedings`
`inproceedings`
`mastersthesis`
`phdthesis`
`manual`
`techreport`
`misc`
`unpublished`

Tabel 8.2 Ufuldstændig liste over felter der kan/skal udfyldes

`author`
`editor`
`title`
`year`
`month`
`journal`
`pages`
`volume`
`edition`
`series`
`number`
`note`
`publisher`
`address`
`url`

Kommandoen `\bibliography`

På det sted hvor litteraturlisten skal være, skriver man kommandoen `\bibliography`, efterfulgt af navnet/navnene på den/de ønskede BibTeX-baser, for eksempel kan man skrive

```
\bibliography{privat,model}
```

for at benytte baserne `privat.bib` og `model.bib`.

Henvisninger og deres udformning

Man henviser til en publikation med kommandoen `\cite`. For eksempel henviser man til de bøger der er vist i Figur 8.1, sådan her:

Goossens et al. (1994)	`\cite{companion}`
Ellegaard Frederiksen (1981)	`\cite{Ellegaard:81}`
Goossens et al. (1994); Ellegaard Frederiksen (1981)	`\cite{companion,Ellegaard:81}`

EKSEMPEL 8.1

Bemærk at man ikke må bruge æ ø å og tilsvarende eksotiske tegn i `\cite` (ligesom man heller ikke må bruge dem i `\label` og `\ref`, jf. side 14).

Brugen af natbib-pakkens `\cite`-kommandoer

Eksempel 8.1 viser hvad der kommer ud af `\cite` når man benytter natbib-pakken (Daly, 1999); hvis man benytter andre (eller ingen) pakker, vil resultatet af `\cite` blive anderledes.

I natbib-pakken findes flere forskellige citations-kommandoer som giver brugeren mulighed for at tilpasse henvisningens udformning efter smag og behag og/eller omstændighederne. De to grundlæggende kommandoer er `\citet` og `\citep`[2] (standardkommandoen `\cite` har nogenlunde samme virkning som `\citet` medmindre man har bedt om noget andet):

EKSEMPEL 8.2

Goossens et al. (1994)	`\cite{companion}`
Goossens et al. (1994)	`\citet{companion}`
(Goossens et al., 1994)	`\citep{companion}`

Man kan angive ét ekstra argument med tekst der skal indsættes efter henvisningen:

EKSEMPEL 8.3

(Goossens et al., 1994, side 10)	`\cite[side 10]{companion}`
Goossens et al. (1994, side 10)	`\citet[side 10]{companion}`
(Goossens et al., 1994, side 10)	`\citep[side 10]{companion}`

Man kan angive to ekstra argumenter hvis der skal indsættes tekst både før og efter henvisningen:

EKSEMPEL 8.4

(se Goossens et al., 1994, side 10)	`\citep[se][side 10]{companion}`
(se Goossens et al., 1994)	`\citep[se][]{companion}`

De »stjernede« udgaver af kommandoerne skriver alle forfatterne:

EKSEMPEL 8.5

Goossens, Mittelbach & Samarin (1994)	`\cite*{companion}`
Goossens, Mittelbach & Samarin (1994)	`\citet*{companion}`
(Goossens, Mittelbach & Samarin, 1994)	`\citep*{companion}`

Man laver multiple henvisninger ved at skrive en liste af citationsnøgler som argument til `\cite`:

EKSEMPEL 8.6

Goossens et al. (1994); Ellegaard Frederiksen (1981)	`\cite{companion,Ellegaard:81}`
Goossens et al. (1994); Ellegaard Frederiksen (1981)	`\citet{companion,Ellegaard:81}`
(Goossens et al., 1994; Ellegaard Frederiksen, 1981)	`\citep{companion,Ellegaard:81}`

Hvis man vil henvise til et årstal uden at angive forfatteren eller omvendt, så kan det også lade sig gøre:

EKSEMPEL 8.7

1994	`\citeyear{companion}`
(1994)	`\citeyearpar{companion}`
Goossens et al.	`\citeauthor{companion}`
Goossens, Mittelbach & Samarin	`\citeauthor*{companion}`

[2] `t` som tekst, `p` som parenteser.

Endvidere kan vi nævne kommandoerne `\citealt` og `\citealp` der virker som `\citet` og `\citep`, bortset fra at alle parenteser bliver udeladt. Endelig kan vi gøre opmærksom på kommandoen `\nocite` hvis effekt er at den tekst der henvises til, kommer med i litteraturlisten; hvis man skriver `\nocite{*}`, så kommer litteraturlisten til at indeholde alt hvad der står i den benyttede BIBTEX-database.

Kald af natbib

Når man bruger IMFUFA-LATEX, anbefales det at aktivere natbib ved at angive `natbib` som option i dokumentklasseerklæringen, altså for eksempel sådan:
`\documentclass[10pt13,natbib]{bog}`
Hvis man vil give options til natbib (se nedenfor), skal man angive dem sammen med dokumentklasse-options'ene, for eksempel sådan:
`\documentclass[10pt13,natbib,super]{bog}`

Hvis man bruger Standard LATEX, skal man være opmærksom på at for at der kan foregå en fredelig sameksistens mellem de to pakker, skal natbib aktiveres *før* babel (jf. Daly, 1999).

Options til natbib

Ved hjælp af forskellige options kan man modificere henvisningernes udseende:

1. `round`, `square` og `curly` ændrer parentesernes udseende,
2. `comma` ændrer skilletegnet fra semikolon til komma,
3. `numbers` ændrer henvisningerne fra forfatter-og-år modellen til [tal] modellen, f.eks. [3].
4. `super` ændrer henvisningerne fra forfatter-og-år modellen til tal-som-toptegn modellen, f.eks. [3],
5. `sort` bevirker at multiple henvisninger bliver skrevet i samme rækkefølge som i litteraturlisten.

   Hvis man har tal-henvisninger kan man bruge `sort&compress` for at få henvisningerne komprimeret (f.eks. 3-5, 7 i stedet for 3, 4, 5, 7).
6. `longnamesfirst` bevirker at første gang en henvisning optræder, kommer alle forfatternavnene med, de følgende gange kommer blot den forkortede liste.

Litteraturlistens udseende

Der kan føres endeløse diskussioner om hvordan en litteraturliste skal udformes, og om hvordan henvisningerne inde i teksten skal se ud. Set fra et LATEX-synspunkt bestemmes litteraturlistens udseende af den valgte *bibliografistil*. Bibliografistilen specificeres ved hjælp af kommandoen `\bibliographystyle` der placeres umiddelbart før `\bibliography`-kommandoen. Der findes en lang række

```
\documentclass[natbib,10pt13,english,danish]{bog}

%% + øvrige præambel-ting

\begin{document}
\maketitle

%% her kommer den egentlige tekst i form af
%% talrige \input{...}-kommandoer

\bibliographystyle{dcu}
\bibliography{typobibl}  % databasen er  typobibl.bib

%% + stikordsregister

\end{document}
```

Figur 8.2 De dele af primærfilen til nærværende tekst som vedrører litteraturliste og -henvisninger.

bibliografistile; her nævnes nogle af dem der er til rådighed i IMFUFAs LaTeX-installation:

- `plainnat` som er en natbib-udgave af standard-stilen `plain`.
- `abbrvnat` som er en komprimeret udgave af `plainnat`.
- `unsrtnat` som er en usorteret udgave af `plainnat`.
- `kluwer` fra forlaget Kluwer.
- `agsm` som er Australian Government publications style.
- `dcu` stammer fra Design Computing Unit, Department of Architecture and Design Science, University of Sidney.

  Nærværende tekst benytter `dcu`, Figur 8.2 viser hvordan.

Bemærk at hvis man ændrer bibliografistilen, så skal man køre LaTeX, BibTeX og endnu engang LaTeX før man kan se virkningen, jf. næste afsnit.

Hvordan gør man?

LaTeX kan ikke selv gøre hele arbejdet med at hente oplysninger fra bibliografi-databasen, der skal assistance fra programmet BibTeX. Det foregår på følgende måde:

1. Når LaTeX sætter teksten, så bliver der i aux-filen skrevet hvilke publikationer der henvises til.
2. Derefter skal man aktivere programmet BibTeX (med primærfilen (eller rettere primærfilens fornavn) som argument). Dette program læser aux-filen, finder de nævnte henvisninger i databasen, og skriver de nødvendige bibliografiske data i en ny hjælpefil der har efternavn `bbl`.

3. Næste gang LaTeX sætter teksten, bliver `bbl`-filen automatisk indlæst og benyttet til at fremstille litteraturlisten.

4. Næste gang igen har `\cite`-kommandoerne adgang til den rigtige information, og alt er nu som det skal være (eller i hvert fald som man har bedt om).

Det ser måske besværligt ud, men det er det ikke; og man behøver kun at køre BibTeX (altså lave en ny `bbl`-fil) når man har indført henvisninger til ikke tidligere citerede publikationer og/eller når man har ændret bibliografistilen.

Om at definere LaTeX-kommandoer

Brugeren kan – for egen regning og risiko – definere egne kommandoer og omdefinere allerede eksisterende kommandoer. Som oftest ønsker man at definitionerne skal være globale, og så bør de stå i præamblen, men de kan faktisk stå hvorsomhelst. Kommandoer defineres med `\newcommand*` og omdefineres med `\renewcommand*`.[3]

Når man vil definere en kommando uden argumenter, er syntaksen

`\newcommand*{`⟨*kommandonavn*⟩`}{`⟨*definition*⟩`}`

Syntaksen for `\renewcommand*` er tilsvarende. Hvis kommandoen skal have argumenter, skrives antallet heraf i kantede parenteser, og i definitionsdelen kan man bruge de formelle argumenter `#1`, `#2` osv. Her følger en række eksempler af forskellig sværhedsgrad:

1. Man kan indføre en kommando som forkortelse for noget tekst der skal bruges mange gange:

```
\newcommand*{\imfufa}{Institut for studiet af matematik
og fysik samt deres funktioner i undervisning,
forskning og anvendelser}
```

2. Man kan omdefinere en eksisterende kommando, for eksempel for at give den et andet typografisk indhold. Her vises hvordan man kan omdefinere kommandoen for de reelle tal (jf. Eksempel 5.35 side 48) så den resulterer i $\mathbb{R}$ i stedet for $\mathbf{R}$:

```
\renewcommand*{\R}{\mathbb{R}}
```

3. Man kan lave (og anvende) kommandoer for at sikre en vis standardisering af den måde som bestemte ting skrives på, eksempelvis personnavne. Her vises en kommando der skriver sit argument i kapitæler:

```
\newcommand*{\person}[1]{\textsc{#1}}
```

[3] Der findes også kommandoerne `\newcommand` og `\renewcommand`; de bruger væsentlig mere af TeX-programmets interne hukommelse og er kun nødvendige i ganske sjældne tilfælde.

Herefter resulterer `\person{K. Gödel}` i K. GÖDEL.

4. LaTeX justerer selv *brøkers* størrelse efter om de står i en fremhævet formel ('display style') eller i en formel inde i teksten ('text style'); ofte har forfatteren andre meninger om den sag, og så kan man spare en del skrivearbejde ved at definere en kommando `\tfrac` der altid laver en 'textstyle' brøk og en kommando `\dfrac` der altid laver en 'displaystyle' brøk:

```
\newcommand*{\tfrac}[2]{{\textstyle\frac{#1}{#2}}}
\newcommand*{\dfrac}[2]{{\displaystyle\frac{#1}{#2}}}
```

5. Hvis man ønsker en standardiseret måde at henvise til figurer og tabeller på (skal man f.eks. skrive ordene figur og tabel med stort eller lille begyndelsesbogstav?), kan man definere kommandoer `\reffigure` og `\reftable` og så *altid* benytte dem når man henviser til figurer og tabeller:

```
\newcommand*{\reffigure}[1]{Figur \ref{#1}}
\newcommand*{\reftable}[1]{Tabel \ref{#1}}
```

Det kan gøres endnu smartere med to af LaTeXs indbyggede navnekonstanter:

```
\newcommand*{\reffigure}[1]{\figurename\ \ref{#1}}
\newcommand*{\reftable}[1]{\tablename\ \ref{#1}}
```

Fidusen ved de sidste definitioner er at de er sproguafhængige; værdierne af navnekonstanterne ændres nemlig automatisk når man skifter sprog (takket være babel-pakken).

6. Når man henviser til nummererede matematiske formler, skriver man traditionelt formlens nummer i parentes og på en sådan måde at nummeret i henvisningen fremtræder på samme måde som i formlen; eksempel: (5.6). Imidlertid er resultatet af LaTeXs `\ref`-kommando et nummer *uden* parenteser, sådan: 5.6. Hvis man vil slippe for selv at skulle (huske at) skrive parenteserne, kan man definere en særlig kommando `\refeqn` som man så *altid* bruger når man henviser til en formel. Kommandoen kan defineres på følgende måde:

```
\newcommand*{\refeqn}[1]{(\ref{#1})}
```

7. Med `enumerate`-konstruktionen får man en liste af nummererede punkter. Det er muligt at omdefinere den måde nummereringen sker på. Her er et eksempel hvor vi omdefinerer nummereringen på niveau 1 (`enumi`) til store romertal med et punktum efter (se også Eksempel 8.12):

I. Punkt 1
II. Punkt 2
III. Punkt 3

```
\renewcommand*{\labelenumi}{%
  \Roman{enumi}.}
\begin{enumerate}
  \item Punkt 1
  \item Punkt 2
  \item Punkt 3
\end{enumerate}
```

EKSEMPEL 8.8

Her er et noget mere avanceret eksempel hvor vi definerer en ny listekonstruktion, en datoliste (som af pladsmæssige grunde er temmelig kort):

Kursusplan:
4.2. Første gang
7.2. Anden gang
11.2. Tredje gang

```
% bør stå i præamblen:
\newenvironment{datoliste}{%
  \renewcommand*{\labelenumi}{%
    \ifcase\arabic{enumi}\or
    4.2.\or 7.2.\or 11.2.\fi}%
  \begin{enumerate}}
  {\end{enumerate}}

% i teksten kan man skrive:
  Kursusplan:
  \begin{datoliste}
    \item Første gang
    \item Anden gang
    \item Tredje gang
  \end{datoliste}
```

EKSEMPEL 8.9

8. Kommandoen `\textqm` sætter anførselstegn » « om sit argument (man kan i øvrigt også benytte `\begin{qmark}` `\end{qmark}`). Hvis man ønsker en anden type anførselstegn, er den rigtige måde at gøre det på at bibeholde kommandoen, men ændre dens typografiske indhold.

   Fra starten er `\textqm` defineret sådan at den har tre forskellige niveauer af anførselstegn:

»Første ›Andet ‘Tredje Niveau’‹«

```
\textqm{Første \textqm{Andet}
    \textqm{Tredje Niveau}}}
```

EKSEMPEL 8.10

Tegnene på de tre niveauer bestemmes af kommandoerne `\leftqmarki`, `\rightqmarki`, `\leftqmarkii`, `\rightqmarkii`, `\leftqmarkiii` og `\rightqmarkiii`, så det er dem der skal omdefineres (helst i præamblen). I eksemplet omdefinerer vi af pladshensyn kun første niveau:

Her er „gåseøjne“ som vi lærte dem i skolen.

```
\renewcommand*{\leftqmarki}{,,}
\renewcommand*{\rightqmarki}{‘‘}
Her er \textqm{gåseøjne}
som vi lærte dem i skolen.
```

EKSEMPEL 8.11

Om tællere

Som det er fremgået, nummererer LaTeX en lang række ting: sider, formler, kapitler, fodnoter osv. Det foregår ved hjælp af nogle interne *tællere* ('counters'); disse har typisk det samme (engelske) navn som det der tælles: `page`, `equation`, `chapter`, `footnote` osv. Som bekendt er et tal ikke det samme som den måde tallet skrives på, heller ikke i LaTeX; der findes kommandoer på brugerniveau der skriver værdien af en tæller enten på den sædvanlige måde (med arabiske tal) eller som store eller små romertal eller som store eller små bogstaver:

EKSEMPEL 8.12

Vi er i Kapitel 8.	`Vi er i Kapitel \arabic{chapter}.`
Vi er i Kapitel VIII.*	`Vi er i Kapitel \Roman{chapter}.`
Vi er i Kapitel viii.†	`Vi er i Kapitel \roman{chapter}.`
Vi er i Kapitel H.‡	`Vi er i Kapitel \Alph{chapter}.`
Vi er i Kapitel h.§	`Vi er i Kapitel \alph{chapter}.`

*kun positive tal
†kun positive tal
‡kun tal fra 1 til 26
§kun tal fra 1 til 26

Til hver tæller hører en intern kommando der leverer tællerens såkaldte *ref-værdi*, dvs. den værdi der faktisk skrives, og som man får som resultat når man henviser (med `\ref`) til det som tælleren tæller. Disse kommandoer hedder `\the...` hvor der på prikkernes plads står navnet på tælleren, f.eks. `\thechapter`, `\thepage` og `\thefootnote`.

EKSEMPEL 8.13

Dette står på side 88.	`Dette står på side \thepage.`

Hvis man ønsker tællerens værdi skrevet på en anden måde, så er fremgangsmåden den at man omdefinerer `\the...`-kommandoen. Hvis man for eksempel ønsker kapitelnumre skrevet med store romertal, så skal man skrive man sådan i præamblen:

```
\renewcommand*{\thechapter}{\Roman{chapter}}
```

Man kan få fodnoter »nummereret« med de særlige fodnotesymboler der ses i Eksempel 8.12, ved at skrive

```
\renewcommand*{\thefootnote}{\fnsymbol{footnote}}
```

Visse objekter nummereres fortløbende, det gælder for eksempel sider og kapitler. Andre objekter, for eksempel fodnoter, nummereres fortløbende inden for kapitler. Atter andre objekter, for eksempel formler, figurer og tabeller (og de indrammede eksempler her i disse noter), nummereres fortløbende inden for kapitler, men deres ref-værdi er af formen ⟨*kapitelnr*⟩.⟨*objektnr*⟩. Eksempelvis er `\thefigure` defineret sådan her[4]:

[4] det er den nu ikke, men dens definition er ækvivalent med den her viste.

```
\newcommand*{\thefigure}{\thechapter.\arabic{figure}}
```

Det betyder at hvis man omdefinerer `\thechapter` til romertal, så nedarves det automatisk til `\thefigure` således at f.eks. den fjerde figur i det tredje kapitel får ref-værdien III.4.

I enkelte tilfælde kan man have brug for at gribe ind i den automatiske nummerering. Man kan selv sætte værdien af en tæller med kommandoen `\setcounter`, for eksempel kan man give tælleren `chapter` værdien 140 ved at skrive

```
\setcounter{chapter}{140}
```

Kapitel 9

Diverse

Kasser og lignende

Rammer – \fbox og \framebox

Man kan indramme ting med kommandoen \fbox:

Her er indrammet tekst, og her er indrammet tekst.	`Her er \fbox{indrammet tekst}, og her er indrammet \fbox{\phantom{usynlig}} tekst.`

EKSEMPEL 9.1

Som det ses, fremstiller kommandoen \phantom *usynlig* tekst.

Størrelsen af en \fbox bestemmes af det der skal indrammes; der findes en mere generel kommando \framebox hvor brugeren selv kan bestemme bredden samt indholdets placering:

En 2 cm lang ramme.	`En \framebox[2cm]{2 cm lang} ramme.`
En 2 cm lang ramme.	`En \framebox[2cm][l]{2 cm lang} ramme.`
En 2 cm lang ramme.	`En \framebox[2cm][r]{2 cm lang} ramme.`
En 2 cm lang ramme.	`En \framebox[2cm][s]{2 cm lang} ramme.`

EKSEMPEL 9.2

Syntaksen for \framebox er

\framebox[⟨*bredde*⟩]{⟨*indhold*⟩}

eller

\framebox[⟨*bredde*⟩][⟨*position*⟩]{⟨*indhold*⟩}.

I det felt hvor man angiver kassens bredde, kan man benytte nogle særlige størrelser, nemlig \width (bredden af ⟨*indhold*⟩), \height (hvor højt ⟨*indhold*⟩

når op over linjen), \depth (hvor langt ⟨*indhold*⟩ når ned under linjen) og \totalheight (⟨*height*⟩+⟨*depth*⟩).

EKSEMPEL 9.3

50% større	`\framebox[1.5\width]{50\% større}`

Kasser uden rammer – \mbox og \makebox

Kommandoerne \mbox og \makebox virker på samme måde som hhv. \fbox og \framebox, blot kommer der ingen ramme om. Alle fire kommandoer kan benyttes overalt, også i formler:

EKSEMPEL 9.4

$x^2 \geq 0$ for alle x	`$x^2 \geq 0 \textrm{ for alle }x$`

(Vedrørende tekst i formler henvises i øvrigt til side 49.)

Kassehævning – \raisebox

Kommandoen \raisebox kan hæve eller sænke en kasse et givet stykke i forhold til linjen; syntaksen er \raisebox{⟨*løft*⟩}{⟨*indhold*⟩}:

EKSEMPEL 9.5

Her er den helt almindelige tekstlinje. Nu hæver vi, og nu sænker vi i forhold til den normale linje, og ser hvad der sker.	`Her er den helt almindelige tekstlinje. Nu \raisebox{2ex}{hæver} vi, og nu \raisebox{-2ex}{sænker} vi i forhold til den normale linje, og ser hvad der sker.`

(Enheden ex, også kaldet *x-højden*, er defineret som højden af bogstavet x i den aktuelle skrift.) LaTeX justerer afstanden sådan at det hævede (sænkede) ikke kolliderer med de nærmeste linjer. Hvis man ikke ønsker dette, kan man angive et eller to valgfrie argumenter der specificerer den ønskede højde og eventuelt også dybde:

EKSEMPEL 9.6

Her er den helt almindelige tekstlinje. Nu hæver vi, og nu sænker vi i forhold til den normale linje, og ser hvad der sker.	`Her er den helt almindelige tekstlinje. Nu \raisebox{2ex}[0ex]{hæver} vi, og nu \raisebox{-2ex}[0ex]{sænker} vi i forhold til den normale linje, og ser hvad der sker.`

\rule

Kommandoen \rule fremstiller et udfyldt rektangel med en given bredde og højde; rektanglet kan hæves/sænkes:

En bjælke: ▬.	`En bjælke: \rule{5mm}{1mm}.`
En løftet bjælke: ▬.	`En løftet bjælke: \rule[-2mm]{5mm}{1mm}.`

EKSEMPEL 9.7

Som bredde eller højde kan man godt angive 0, eller rettere 0 af en eller anden længdeenhed. Så får man et tekstobjekt som i sig selv er usynligt, men som godt kan have en synlig virkning, for eksempel kan man få en kasse til at blive større:

X X	`\fbox{X} \fbox{X\rule{0ex}{3ex}}`

EKSEMPEL 9.8

En `rule` med bredde 0 hedder en *strut*, og der findes ligefrem en LaTeX-kommando `\strut`, der svarer til
`\rule[-.3\baselineskip]{0pt}{\baselineskip}`.

Strutter kan være nyttige hvis man vil fremtvinge parenteser af forskellig størrelse, for eksempel for at lette læseligheden af en matematisk formel:

$(a-(c+d))^2$	`$(a-(c+d))^2$`
$(a-(c+d))^2$	`$\left(a-(c+d)\right)^2$`
$\left(a-(c+d)\right)^2$	`$\left(\strut a-(c+d)\right)^2$`

EKSEMPEL 9.9

Forklaring: De parenteser der kommer med `\left` `\right` (se side 45), får en størrelse der afhænger af højden af det der står mellem parenteserne; i anden linje i eksemplet er »mellemteksten« ikke høj nok til at parenteserne forstørres, derfor må man indsætte en strut (her med `\strut`).

\descriptionlabelfont

I `description`-konstruktioner bliver de enkelte punkters 'label' sat med fed skrift, se Eksempel 2.13. Undertiden kan man ønske sig en anden skrifttype end fed, og et sådant ønske lader sig let effektuere i IMFUFA-LaTeX. Hvis man skriver en kommando af formen `\descriptionlabelfont{`⟨*fontkommando*⟩`}`, så bliver punkternes 'labels' sat med den font der svarer til ⟨*fontkommando*⟩. Her er et eksempel hvor man vælger kursiv; ændringen vil gælde alle følgende `description`-konstruktioner:

Gæs spises Mortens aften, ihukommende Skt. Martin.
Ænder spises til jul, fordi en gås er for stor.
Høns spises i asparges.
	`\descriptionlabelfont{\itshape}`
	`\begin{description}`
	`\item[Gæs] spises Mortens aften, ihukommende Skt. Martin.`
	`\item[Ænder] spises til jul, fordi en gås er for stor.`
	`\item[Høns] spises i asparges.`
	`\end{description}`

EKSEMPEL 9.10

Dernæst et eksempel hvor man vælger normalskriften; denne gang gælder ændringen kun den aktuelle `description`-konstruktion (fordi ændringen foretages inden i konstruktionen):

EKSEMPEL 9.11

Gæs spises Mortens aften, ihukommende Skt. Martin.
Ænder spises til jul, fordi en gås er for stor.
Høns spises i asparges.
	`\begin{description}`
	`\descriptionlabelfont{}`
	`\item[Gæs] spises Mortens aften, ihukommende Skt. Martin.`
	`\item[Ænder] spises til jul, fordi en gås er for stor.`
	`\item[Høns] spises i asparges.`
	`\end{description}`

Overspring tekst

Hvordan beder man LaTeX om at springe et vist tekstafsnit over? En mulighed er at sætte kommentartegn (%) i begyndelsen af hver linje, men det er besværligt ved større tekstafsnit. Det er lettere og elegantere at gøre som vist her:

EKSEMPEL 9.12

Her er noget tekst. Her er noget mere tekst.	`Her er noget tekst.`
	`\iffalse`
	`Denne del af teksten springes over.`
	`\fi`
	`Her er noget mere tekst.`

Ideen er at når LaTeX møder en konstruktion af formen `\if`⟨*logisk udtryk*⟩ ⟨*tekst*⟩ `\fi`, så overspringes ⟨*tekst*⟩ medmindre ⟨*logisk udtryk*⟩ har værdien `true` (og i eksemplet er ⟨*logisk udtryk*⟩ den logiske konstant `false`).

`\verb` og `verbatim`

Læseren har måske spekuleret over hvordan man egentlig bærer sig ad med at skrive den tekst der viser LaTeX-kommandoerne. Svaret er at det gøres med `\verb`- og `\verb*`-kommandoerne og med `verbatim`- og `verbatim*`-konstruktionerne.

`\verb` og `\verb*`

Man bruger `\verb` og `\verb*` til ting der skal stå inde i teksten:

EKSEMPEL 9.13

Her er nogle tegn: 230\ ?%& # abc	`Her er nogle tegn: \verb"230\ ?%& # abc"`
Her er nogle tegn: 230\␣?%&␣#␣abc	`Her er nogle tegn: \verb*"230\ ?%& # abc"`

Kommandoen \verb virker på følgende måde: Først ser den hvad det er for et tegn der står umiddelbart efter kommandoen; lad os kalde dette tegn for *kontroltegnet.* I eksemplet er kontroltegnet et ", men det kan være stort set hvad som helst, dog ikke et bogstav[1] eller en *. Derefter sker der en nøjagtig kopiering af alle de efterfølgende tegn, sluttende umiddelbart før den næste forekomst af kontroltegnet. Kommandoen \verb* virker på samme måde, blot skriver den mellemrum som ␣.

Bemærk at der ikke må være linjeskift i argumentet til \verb eller \verb*. Bemærk også at \verb og \verb* ikke må optræde inde i argumentet til en anden kommando (de kan således ikke bruges i fodnoter eller i kapitel- eller afsnitsoverskrifter).

Hvis man har brug for at skrive \verb mange gange, kan man eventuelt benytte kommandoen \MakeShortVerb (fra pakken shortvrb):

Kommandoen "- har en særlig betydning. Kender du den?	`\MakeShortVerb{\?}`
	`Kommandoen ?"-? har en`
	`særlig betydning.`
	`\DeleteShortVerb{\?}`
	`Kender du den?`

EKSEMPEL 9.14

I eksemplet tildeles tegnet ? en specialbetydning sådan at man kan skrive ?⟨*tegn*⟩? i stedet for f.eks. \verb"⟨*tegn*⟩". Dette benyttes til at skrive "-. Derefter ophæves den særlige betydning af ?, og dette tegn kan nu bruges i sin normale betydning. – Det »særlige tegn« skal helst være et man ikke bruger ret meget i den almindelig tekst, så ? er måske et dårligt valg; tit er | et godt valg.

verbatim og verbatim*

Tekst der omgives af \begin{verbatim} \end{verbatim} kopieres nøjagtigt; hvis man erstatter verbatim med verbatim* bliver mellemrum gengivet som ␣.

Her er tekst.
 	
`f"%!?** =&\/`
`\endinput`
`abcæøå`
	`Her er tekst.`
	`\begin{verbatim}`
	`  f"%!?** =&\/`
	`  \endinput`
	`  abcæøå`
	`\end{verbatim}`

EKSEMPEL 9.15

Sådanne konstruktioner må ikke optræde i argumentet til en anden kommando.

Om filnavne

De forskellige styresystemer til computere har lidt forskellige regler for filnavne. Hvis man vil kunne bruge sine TEX-filer i »alle« styresystemer, skal man navngive dem efter 8.3-systemet, dvs. et fornavn på højst otte tegn og et efternavn

[1] Hvis kontroltegnet var et bogstav, så ville det jo blive opfattet som hørende med til kommandonavnet som så ikke ville være \verb.

på højst tre. Tegnene kan være bogstaver (a–z) og cifre og enkelte andre tegn; disse »enkelte andre tegn« må *absolut ikke* være tegn som / . \ * ? | < > De må heller ikke være æ ø å (selv om nogle styresystemer tillader filnavne med æ ø å). Med hensyn til bogstaverne så er der ikke forskel på store og små bogstaver.

Bemærk venligst: Nogle styresystemer bruger en backslash (\) som skilletegn mellem de enkelte biblioteksnavne der indgår i det fulde filnavn, andre en forwardslash (/). Men når man skriver filnavnet inde i LaTeX, skal man altid bruge forwardslash. Eksempel: Hvis man vil indlæse en fil som efter styresystemets syntaks hedder `c:\modul3\projekt\kapitel2.tex`, så skal man skrive `\input{c:/modul3/projekt/kapitel2.tex}`.

Fremmede tex-filer, \inputencoding

Hvis man udelukkende arbejder med én slags computere, f.eks. udelukkende med DOS-computere eller udelukkende med UNIX-computere, så er der en del problemer man slipper for, blandt andet problemer der kommer af at der ikke er en fælles standard for hvordan europæiske-men-ikke-amerikanske bogstaver (som for eksempel æ ø é ü å) skal kodes internt i maskinen. Dette afsnit handler om den type problemer.

Først viser vi hvordan problemet ytrer sig. Lad os sige at man har benyttet en MS-Windows-baseret editor til at fremstille en tex-fil indeholdende ordet `forårsjævndøgn`, og at man derefter flytter denne fil over til en DOS-baseret editor. Så er ordet pludselig blevet til `forÕrsjµvnd°gn`, og når man putter det ind i LaTeX, bliver resultatet måske heller ikke godt: Der kommer nogle fejlmeddelelser, og slutresultatet ser sådan ud:

EKSEMPEL 9.16

forÕrsj*µvnd°gn*	`forÕrsjµvnd°gn`

Løsningen på problemet er at benytte kommandoen `\inputencoding`[2] der fortæller LaTeX hvordan input-teksten skal oversættes til intern LaTeX-kode:

EKSEMPEL 9.17

forårsjævndøgn	`forårsjævndøgn`
	`\inputencoding{ansinew}`
forårsjævndøgn	`forÕrsjµvnd°gn`

Her er de `\inputencoding`-værdier som skønnes at være af størst interesse for

[2]Kommandoen `\inputencoding` er defineret i pakken inputenc der automatisk aktiveres af IMFUFA-LaTeX.

den typiske LaTeX-bruger i Danmark:

`ansinew`	til LaTeX på MS-Windows-maskiner
`applemac`	til LaTeX på Apple Macintosh
`ascii`	til LaTeX på en generel maskine
`cp850`	til LaTeX på DOS-maskiner
`cp865`	til LaTeX på DOS-maskiner
`latin1`	til LaTeX på UNIX-maskiner

IMFUFA-LaTeX vil normalt være installeret sådan at den af sig selv benytter en hensigtsmæssig værdi.

Det kan i øvrigt nævnes at visse tekst-editorer er så avancerede at de kan gemme tekst i flere forskellige formater.

Kapitel 10

Forskellige oplysninger

Hvor kommer LaTeX fra?

Den altafgørende hovedperson i TeXs tilblivelse er matematikeren og datalogen *Donald E. Knuth* fra Stanford University. Han påbegyndte i 1977 et stort projekt gående ud på at udvikle et tekstbehandlingssystem, TeX[1], og et tilhørende system, METAFONT, til at beskrive og generere skriftsnit; ved hjælp af METAFONT skabtes Computer Modern skrifterne. I 1986 var TeX-projektet stort set tilendebragt hvilket markeredes med fembindsværket *Computers and Typesetting* med den fuldstændige beskrivelse af TeX, METAFONT og Computer Modern, og i 1990 bekendtgjorde Knuth at 'my work on developing TeX, METAFONT, and Computer Modern has come to an end. I will make no further changes except to correct extremely serious bugs'.

I begyndelsen af 1980erne udviklede *Leslie Lamport* LaTeX som en overbygning på TeX (Lamport, 1985, 1994). I 1989 indledtes det såkaldte LaTeX3-projekt hvis formål kort opsummeres til at 'it will greatly increase the range of documents which can be processed; and it will provide a flexible interface for typographic designers to easily specify the formatting of a class of documents'; LaTeX3-projektgruppen består af personer fra en lang række lande, især europæiske. I forbindelse med LaTeX3-projektet er der udviklet forskellige pakker, som for eksempel babel, og der er udviklet det nye 256-tegns TeX-tegnsæt og de tilhørende European Computer Modern skriftsnit. – Vi må henvise til Goossens et al. (1994, 1997) og henvisninger heri for mere detaljerede oplysninger om TeX og LaTeX; hvis man *køber* de nævnte bøger er man i øvrigt med til at støtte LaTeX3-projektet økonomisk.

TeX og METAFONT er generelle systemer, og for at de kan benyttes konkret, skal de tilpasses, implementeres, til den givne type datamater med det givne operativsystem; der findes TeX-udgaver til alle gængse datamater/operativsystemer, jf. Tabel 10.1.

De fleste TeX-brugere sætter pris på en brugerflade der er lidt mere avanceret end blot en kommandolinje; brugerfladen kan typisk være integreret i en teksteditor. Der findes brugbare *TeXShells* til alle slags operativsystemer.

[1] Bogstaverne T E X er græske, nemlig stort Tau Epsilon Khi.

Tabel 10.1 Free or shareware TEX implementations (fra Rahtz et al., 1999)

TeX Live is a CD-ROM package from TUG and other user groups which contains setups for many Unix systems, Win32, and Amiga.

Web2c, Karl Berry's monumental and widely-used system for compiling all the important TEX software in C. Now maintained by Olaf Weber.

teTeX, Thomas Esser's packaging of Web2c for many Unix systems.

NTeX (Linux TeX based on Web2c 7.2)

OzTeX for the Macintosh, by Andrew Trevorrow.

CMacTeX for Macintosh, by Tom Kiffe.

TeXgX, a Macintosh TEX (shareware) by Jonathan Kew that can access the sophisticated features of GX fonts (such as automatic ligatures, contextual swashes, etc).

Gary Gray maintains a full Web page for **Macintosh** TEX and LATEX stuff.

MikTeX, a highly-regarded setup for Windows 95/NT.

Web2c-win32 (fpTeX) a complete port of Web2c for Win32 by Fabrice Popineau, including a port of Xdvi for windows (windvi).

OS/2 TeX, shareware setup for OS/2 put together by Jürgen Kleinböhl.

emTeX, the classic DOS and OS/2 TEX setup by Eberhard Mattes.

Don Knuth har skænket TEX til verdenssamfundet, eller sagt på en anden måde: TEX er 'public domain' software som alle frit må benytte og som ingen må tage penge for (ud over hvad der dækker omkostninger til kopiering o.lgn.). TEX og alt hvad dertil hører kan hentes via internettet fra det såkaldte »Comprehensive TEX Archive Network« (CTAN) der er et netværk af samarbejdende ftp-arkiver; for en nordeuropæer er de relevante adresser `ftp.dante.de` og `ftp.tex.ac.uk`. IMFUFA-LATEX findes på `dirac.ruc.dk/imfufalatex`.

Noter til teksten

Her er lidt baggrundsoplysninger og kildehenvisninger til nogle af eksemplerne i teksten.

Side 16: Teksten i Figur 2.1 og 2.2 er fra Steen Steensen Blichers novelle *En Landsbydegns Dagbog*, en af perlerne i dansk litteratur.

Der er citeret efter Knud Sørensens udvalg af Blicher-noveller, udsendt på Gyldendal 1995.

Side 23: De latinske ord *Sola scriptura* betyder *Skriften alene.* – Vendingen benyttes ofte som et slagord til at udtrykke en bestemt side af Luthers reformatoriske virke, nemlig at det er skriften, dvs. teksten, dvs. Biblen, alene, og ikke traditionen eller (kirke)institutionen der betyder noget.

Side 24: *Quod scripsi, scripsi* er latin og betyder *Hvad jeg skrev, det skrev jeg*, og det kan enhver forfatter føle trang til at fremføre, stillet over for besværlige læseres brokkerier; således også Pilatus da han havde forfattet indskriften over Jesu kors (Johs. 19,22).

Side 29 og 66: Robert Musil (1880-1942), østrigsk forfatter, uddannet som ingeniør, arbejdet med videnskabsteori og filosofi. Hans hovedværk er den store ufuldendte roman *Manden uden egenskaber* (Der Mann ohne Eigenschaften) der foregår i Wien i begyndelsen af århundredet. I øvrigt er titelpersonen der synes at have mange egenskaber til fælles med forfatteren, faktisk matematiker.

Romanen er oversat til dansk af Karsten Sand Iversen og udkommer hos Gyldendal. Citatet er fra side 45-46.

Side 32: Visen i Eksempel 4.16 er en »Vise fra Fyn« ifølge Højskolesangbogen.

Side 37: Teksten i Eksempel 4.17 er fra Kapitel 2 i Arthur Conan Doyles *A Study In Scarlett – The First Book About Sherlock Holmes.*

Tillæg A

Kort oversigt over IMFUFA-LaTeX

Dette er en yderst kortfattet oversigt over IMFUFA-LaTeX. De fleste af de nævnte faciliteter er omtalt nærmere andetsteds i nærværende publikation.

1. IMFUFA-LaTeX består blandt andet af dokumentklasserne bog, rapport, notat, artikel og brev.

2. Dokumentklasserne accepterer blandt andet følgende options:

   - `danish`, `english`, `german`
   - `10pt13` (foruden de sædvanlige `10pt`, `11pt` og `12pt`)
   - `imfufapar`
   - `ccfonts` Concrete-skriften plus layout-mæssige småændringer
   - `natbib` indlæser natbib *før* babel (som altid indlæses)
   - `cp850`, `ascii`, `applemac`, `ansinew`, `latin1` input encodings

3. Der defineres blandt andet følgende kommandoer til brug inde i teksten:

   - `\textqm` anførselstegn
   - `\bpar` 'big `\par`'
   - `\nicefrac` og `\oldstylenums` forbedrede udgaver der bruger samme skrift som omgivelserne
   - `\emptypages`
   - `theorem` og `proof`
   - `\promille` `\celsius` `\musicalnote` `\euro` `\numero`
   - `"/` en skråstreg der virker rigtigt ved orddeling (til ordsammensætninger som f.eks. import/eksport). Kun i danskprogede tekster.
   - `\-` standardkommandoen `\-` omdefineres sådan at orddeling er tilladt på begge sider af `\-`. Kun i danskprogede tekster.

4. Der defineres blandt andet følgende kommandoer til definitioner og layoutmæssige ting:
   - `\newproof`
   - `\RaggedRight` (bedre end standardkommandoen `\raggedright`. En kommando med samme navn og formål findes i (Schröder, 1996))
   - `lheadings` en ny 'pagestyle' i dokumentklasserne bog og rapport
   - `\descriptionlabelfont`
   - `\captionbodyfont`
   - `\captionnumberfont`
5. Der defineres blandt andet følgende matematikkommandoer:
   - `\N`, `\Z`, `\Q`, `\R`, `\C`
   - omdefinering af `\Re` og `\Im`
   - `\Exp`, `\Var`, omdefinering af `\Pr`
   - `\binom` (AMS-klasserne har en kommando med samme navn og formål)
   - `\ntup`
   - `\ton` spejlvendt `\not`
   - `\eps` (ε)
   - `\limarrow` så man kan skrive $x_n \xrightarrow[n\to\infty]{} 1$
6. Der defineres blandt andet følgende vedr. skriftvalg:
   - pakken dunhill definerer kommandoen `\textdunh` til valg af Dunhill-skriften (den tilsvarende erklæring er `\dunhfamily`)
   - pakken fibonac definerer kommandoen `\textfib` til valg af Fibonacci-skriften (den tilsvarende erklæring er `\fibfamily`)
   - pakken boldlght definerer kommandoerne `\textbl` og `\textbc` der leverer hhv. en **bold (non-extended)** og en **semibold condensed** skrift (de tilsvarende erklæringer er `\blseries` hhv. `\bcseries`).
   - `\textco` kommando til valg af 'text companion'[1] (den tilsvarende erklæring er `\tcfont`)
7. Andet:
   - Stikordsregister og litteraturliste kommer automatisk med i indholdsfortegnelsen

[1] Et supplerende tekst-tegnsæt der bl.a. indeholder de gammeldags tal (1234567890) og symboler som €, ‰, °C, ♪ og №.

Litteratur

Braams, J. (1999). Babel, a multilingual package for use with LaTeX's standard document classes, Findes på CTAN.

Carlisle, D. (1999). The bm package, Findes på CTAN. Part of the Standard LaTeX 'Tools Bundle'.

Carlisle, D. P. & Rahtz, S. P. Q. (1999). The graphicx package, Findes på CTAN. Part of the Standard LaTeX 'Graphics Bundle'.

Dahl, S. (1944). *Forsøg til en Ordbog for Bogsamlere*, Povl Branners Forlag, København.

Daly, P. W. (1999). Natural sciences citations and references (author–year and numerical schemes), Findes på CTAN.

Ellegaard Frederiksen, E. (1981). *Bogens Funktion og Æstetik*, Christian Ejlers' Forlag, København.

Goossens, M., Mittelbach, F. & Samarin, A. (1994). *The LaTeX Companion*, Addison-Wesley, Reading, Massachusetts.

Goossens, M., Rahtz, S. & Mittelbach, F. (1997). *The LaTeX Graphics Companion: Illustrating documents with TeX and PostScript*, Addison-Wesley, Reading, Massachusetts.

Lamport, L. (1985). *LaTeX – A Document Preparation System*, Addison-Wesley Publishing Company, Reading, Massachusetts.

Lamport, L. (1994). *LaTeX – A Document Preparation System*, second edn, Addison-Wesley Publishing Company, Reading, Massachusetts.

Mittelbach, F. (1989-95). An extension of the LaTeX theorem environment, Findes på CTAN. Part of the 'tools' bundle of standard LaTeX packages.

Rahtz, S., Goossens, M. & Beeton, B. (1999). Free or shareware TeX implementations. List compiled for the TeX Users Group.
URL: *www.tug.org/interest.html#free*

Reimer, E. (1990). *Det typografiske system*, Forening for Boghaandværk, København.

Schmidt, W. (1999). The LaTeX 2_ε package `ccfonts`, Findes på CTAN.

Schröder, M. (1996). The ragged2e-package, Findes på CTAN.

Stikord